

# **PREDICTIVE ENERGY MANAGEMENT STRATEGY FOR A PARALLEL HYBRID ELECTRIC VEHICLE**

A Thesis  
Presented to  
The Academic Faculty

by

Sriganesh Sriram

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science in the  
School of Electrical and Computer Engineering

Georgia Institute of Technology  
December 2016

Copyright © 2016 by Sriganesh Sriram

# PREDICTIVE ENERGY MANAGEMENT STRATEGY FOR A PARALLEL HYBRID ELECTRIC VEHICLE

Approved by:

Dr. David G Taylor, Advisor  
School of Electrical and Computer  
Engineering  
*Georgia Institute of Technology*

Dr. Magnus Egerstedt  
School of Electrical and Computer  
Engineering  
*Georgia Institute of Technology*

Dr. Yorai Wardi  
School of Electrical and Computer  
Engineering  
*Georgia Institute of Technology*

Dr. Michael Leamy  
School of Mechanical Engineering  
*Georgia Institute of Technology*

Date Approved: 26th August 2016

*To my wife, Athithi*  
*and my mother Rajeswari.*

## ACKNOWLEDGEMENTS

I would firstly like to thank my advisor Dr. David Taylor for the immense support he provided throughout my graduate studies and for giving me a funded leadership position on the Georgia Tech EcoCAR3 team. I am also very grateful for the confidence he had in me, for the words of encouragement and for giving me a free hand in exploring this topic. The numerous technical discussions we had were very educative, interesting and more importantly, a lot of fun.

I am also extremely grateful to the other faculty advisors for the Georgia Tech EcoCAR3 team, Dr. Michael Leamy and Dr. Tom Fuller, for their support over the last two years.

Being a part of the EcoCAR3 team was a very fruitful and enjoyable experience. I would like to thank all the colleagues I worked with for their support and for the great memories.

I would also like to thank Dr. Magnus Egerstedt and Dr. Yorai Wardi for agreeing to be a part of the thesis committee.

Finally I would also like to express my gratitude to my family for their love and support during the period of my graduate studies.

# TABLE OF CONTENTS

<b>DEDICATION</b>	<b>iii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iv</b>
<b>LIST OF TABLES</b>	<b>viii</b>
<b>LIST OF FIGURES</b>	<b>x</b>
<b>SUMMARY</b>	<b>xiv</b>
<b>I INTRODUCTION</b>	<b>1</b>
1.1 Motivation	2
1.2 Structure of the Thesis	3
<b>II VEHICLE MODEL FOR A PARALLEL HEV</b>	<b>4</b>
2.1 Chassis Model	5
2.2 Transmission and Torque Converter	7
2.3 Internal Combustion Engine	9
2.4 Electric Machine	12
2.5 High Voltage Battery Pack	15
2.6 Overall Vehicle Plant Model Overview	18
2.7 Drive Cycle and Driver Torque Request	19
<b>III ROUTE SELECTION AND DERIVATION OF ROAD GRADIENT</b>	<b>21</b>
3.1 Route Selection	21
3.2 Extraction of Road Gradient	24
<b>IV BASIC SUPERVISORY CONTROLLER</b>	<b>28</b>
4.1 Axle Torque Request	29
4.2 Plant Input	31
4.3 Charge Depletion Mode	31
4.4 Charge Sustain Mode	34

4.5	Development of State Machine . . . . .	36
4.6	Charge Depletion Mode Simulation . . . . .	43
4.7	Charge Sustain Mode Simulation . . . . .	44
<b>V</b>	<b>EQUIVALENT CONSUMPTION MINIMIZATION STRATEGY</b>	<b>48</b>
5.1	ECMS Overview . . . . .	49
5.1.1	Equivalence Factor . . . . .	49
5.1.2	Control Strategy . . . . .	50
5.2	Simulink Structure for ECMS Supervisory Control . . . . .	55
5.3	Simulation Results . . . . .	64
5.3.1	Comparison of ECMS to Basic Supervisory Controller . . . . .	64
5.3.2	Consideration of Real World Routes . . . . .	71
5.4	Conclusion . . . . .	75
<b>VI</b>	<b>PREDICTIVE CONTROL . . . . .</b>	<b>77</b>
6.1	Method . . . . .	78
6.2	Simulation of Navigation Device . . . . .	81
6.3	Simulation Results . . . . .	82
6.4	Conclusion . . . . .	89
<b>VII</b>	<b>SUMMARY AND CONCLUSIONS . . . . .</b>	<b>90</b>
<b>APPENDIX A</b>	<b>— TRANSMISSION CONTROL MODULE . . . . .</b>	<b>92</b>
<b>APPENDIX B</b>	<b>— ENGINE CONTROL MODULE . . . . .</b>	<b>95</b>
<b>APPENDIX C</b>	<b>— DRIVE CYCLES . . . . .</b>	<b>97</b>
<b>APPENDIX D</b>	<b>— EQUIVALENCE FACTOR AND POWER THRESH- OLD CALCULATIONS FOR ECMS . . . . .</b>	<b>100</b>
<b>APPENDIX E</b>	<b>— SIMULINK IMPLEMENTATION FOR ECMS POWER LIMIT CALCULATIONS . . . . .</b>	<b>102</b>
<b>APPENDIX F</b>	<b>— CALCULATION OF EQUIVALENT GASOLINE FUEL CONSUMPTION . . . . .</b>	<b>106</b>
<b>APPENDIX G</b>	<b>— PRSG PROBLEM FORMULATION . . . . .</b>	<b>108</b>

APPENDIX H — MATLAB CODE FOR THE PRSG . . . . .	111
REFERENCES . . . . .	119

## LIST OF TABLES

1	Tire Parameters . . . . .	6
2	Gearbox Ratios . . . . .	8
3	Gearbox Inertias . . . . .	8
4	Vehicle Parameters . . . . .	19
5	Basic Supervisory Controller Powertrain States . . . . .	43
6	Parameters for Causal Feedback Control . . . . .	55
7	ECMS Powertrain States . . . . .	65
8	E85 Fuel Consumption Comparison between Basic Supervisory Controller and ECMS for US06 City and US06 Highway Drive Cycles . .	69
9	E85 Fuel Consumption Comparison between Basic Supervisory Controller and ECMS for HWFET and US505 Drive Cycles . . . . .	69
10	Weighted E85 Fuel Consumption (kg/100km) Comparison for Basic Supervisory Controller and ECMS . . . . .	69
11	Weighted E85 Fuel Consumption (L/100km) Comparison for Basic Supervisory Controller and ECMS . . . . .	70
12	Equivalent Gasoline Fuel Consumption (L/100km) Comparison for Basic Supervisory Controller and ECMS . . . . .	70
13	Fuel Economy (miles/gallon) Gasoline Comparison for Basic Supervisory Controller and ECMS . . . . .	70
14	Percentage Improvement in Fuel Economy for ECMS over Basic Supervisory Controller . . . . .	70
15	E85 Fuel Consumption and Change in SOC for US06 City and US06 Highway Drive Cycles for ECMS . . . . .	74
16	E85 Fuel Consumption and Change in SOC for HWFET and US505 Drive Cycles for ECMS . . . . .	74
17	Weighted E85 Fuel Consumption ECMS . . . . .	74
18	Equivalent Gasoline Fuel Consumption for ECMS . . . . .	75
19	E85 Fuel Consumption Comparison for US06 City and US06 Highway Drive Cycles for Basic Supervisory Controller, ECMS and pRSG . . .	87
20	E85 Fuel Consumption and SOC Change Comparison for HWFET and US505 Drive Cycles for Basic Supervisory Controller, ECMS and pRSG	88



21	Weighted E85 Fuel Consumption (kg/100km) Comparison for Basic Supervisory Controller, ECMS and pRSG . . . . .	88
22	Weighted E85 Fuel Consumption (L/100km) Comparison for Basic Supervisory Controller, ECMS and pRSG . . . . .	88
23	Equivalent Gasoline Fuel Consumption (L/100km) Comparison for Basic Supervisory Controller, ECMS and pRSG . . . . .	88
24	Fuel Economy (miles/gallon) Gasoline Comparison for Basic Supervisory Controller, ECMS and pRSG . . . . .	88
25	Percentage Improvement in Fuel Economy for pRSG-ECMS over ECMS	89
26	Parameters for the pRSG-ECMS . . . . .	109

## LIST OF FIGURES

1	GT ECoCAR3 Vehicle Architecture . . . . .	5
2	Model for tires and Brakes . . . . .	6
3	Chassis Model . . . . .	7
4	Transmission Model . . . . .	9
5	Throttle Body Model . . . . .	10
6	Air Intake Manifold Model . . . . .	10
7	Core Engine Model . . . . .	11
8	Overall Engine Model . . . . .	12
9	Engine Torque and Power . . . . .	12
10	Calculation of EM Torque . . . . .	14
11	EM Torque vs Speed . . . . .	14
12	Calculation of EM Efficiency . . . . .	14
13	Calculation of EM Power and Current . . . . .	15
14	Calculation of Battery SOC . . . . .	16
15	Calculation of Battery OCV . . . . .	16
16	Calculation of Battery Internal Resistance . . . . .	16
17	Calculation of Battery Voltage . . . . .	17
18	Thevenin's Equivalent Circuit for Battery Model . . . . .	17
19	Overall Vehicle Plant Model . . . . .	19
20	Drive Cycle and Driver Torque Request . . . . .	20
21	Route from Zamenhofstrasse to Wernerstrasse . . . . .	22
22	Route from UW to Space Needle in Seattle . . . . .	22
23	Route in Gatlinburg . . . . .	23
24	Route in Sausalito California . . . . .	23
25	Result for Elevation Request to Google Maps Elevation API . . . . .	25
26	Elevation for Stuttgart Route . . . . .	26
27	Elevation for Seattle Route . . . . .	26

28	Elevation for Gatlinburg Route . . . . .	27
29	Elevation for Sausalito Route . . . . .	27
30	ICE and EM Maximum Torque . . . . .	29
31	EM Maximum Torque Limit . . . . .	30
32	EM Fade Off Factors . . . . .	30
33	ICE and EM Components of Max Axle Torque . . . . .	32
34	Coasting and CD States . . . . .	39
35	Coasting and CS Transition States . . . . .	41
36	Coasting and Braking States . . . . .	42
37	CD Mode Desired Speed and Vehicle Speed . . . . .	44
38	CD Mode Engine and Motor Speed . . . . .	44
39	CD Mode Engine and Motor Torque . . . . .	45
40	CD Mode Powertrain State, Battery SOC and Gear Number . . . . .	45
41	CS Mode Desired Speed and Vehicle Speed . . . . .	45
42	CS Mode Engine and Motor Speed . . . . .	46
43	CS Mode Engine and Motor Torque . . . . .	46
44	CS Mode Powertrain State, Battery SOC and Gear Number . . . . .	46
45	ECMS Structure . . . . .	50
46	Simplified Vehicle Plant Model . . . . .	51
47	Optimal Control Regions at Gear 1 when Equivalence Factor is 3.5 . . . . .	54
48	Optimal Control Regions at Gear 1 when Equivalence Factor is 4.05 . . . . .	54
49	Causal Feedback Controller . . . . .	55
50	Component Limits Block . . . . .	56
51	Component Limits Calculations . . . . .	56
52	Component Efficiency . . . . .	57
53	Calculation of Equivalence Factor . . . . .	57
54	Overview of Power Threshold Calculations for Mode Transitions . . . . .	58
55	Overview of Power Threshold Selection for Current Mode . . . . .	59
56	Calculations of Power Threshold Selection for Current Mode . . . . .	59

57	Truth Table Ports . . . . .	60
58	Truth Table Conditions . . . . .	60
59	Truth Table Actions . . . . .	61
60	ECMS State Machine Part 1 . . . . .	62
61	ECMS State Machine Part 2 . . . . .	62
62	ECMS State Machine Part 3 . . . . .	63
63	Power to Torque Conversion . . . . .	63
64	SOC Plot for ECMS Simulation for US505 Drive Cycle . . . . .	65
65	Powertrain State Plot for ECMS Simulation for US505 Drive Cycle . . . . .	66
66	SOC Plots for BSC and ECMS when Simulated for US06 City Drive Cycle . . . . .	66
67	SOC Plots for BSC and ECMS when Simulated for US06 Highway Drive Cycle . . . . .	67
68	SOC Plots for BSC and ECMS when Simulated for HWFET Drive Cycle . . . . .	67
69	SOC Plots for BSC and ECMS when Simulated for US505 Drive Cycle . . . . .	68
70	SOC Plot for ECMS Simulation for US06 City Drive Cycle and for Stuttgart Route . . . . .	72
71	SOC Plot for ECMS Simulation and for US06 Highway Drive Cycle and for Sausalito Route . . . . .	72
72	SOC Plot for ECMS Simulation for HWFET Drive Cycle and for Gatlinburg Route . . . . .	73
73	SOC Plot for ECMS Simulation for US505 Drive Cycle and for Seattle Route . . . . .	73
74	Block Diagram of pRSG - ECMS Algorithm . . . . .	78
75	Structure of pRSG-ECMS . . . . .	81
76	Reference SOC and Road Elevation for Stuttgart Route . . . . .	83
77	Reference SOC and Road Elevation for Seattle Route . . . . .	84
78	Reference SOC and Road Elevation for Sausalito Route . . . . .	84
79	Reference SOC and Road Elevation for Gatlinburg Route . . . . .	85
80	SOC Plots for Simulation of US505 Cycle in Seattle Route . . . . .	85
81	SOC Plots for Simulation of US06 City Cycle in Stuttgart Route . . . . .	86

82	SOC Plots for Simulation of US06 Highway Cycle in Sausalito Route	86
83	SOC Plots for Simulation of HWFET Cycle in Gatlinburg Route . . .	87
84	Transmission Up-shift and Down-shift Threshold Curves . . . . .	92
85	TCM State Model . . . . .	93
86	Engine Control State Machine . . . . .	95
87	US06 City Drive Cycle . . . . .	97
88	US06 Highway Drive Cycle . . . . .	98
89	HWFET Drive Cycle . . . . .	98
90	US505 Drive Cycle . . . . .	99
91	ECMS Power Limit - Inputs . . . . .	102
92	ECMS Power Limit Calculations - $s_{Boost}$ , $s_{Gen}$ and $P_{EM}^{bo}$ . . . . .	103
93	ECMS Power Limit Calculations - $P_{EM}^{re}$ and $\frac{s}{\eta} - \frac{1}{e}$ . . . . .	103
94	ECMS Power Limit Calculations - $\frac{4s\beta P_0}{\eta^2 e}$ and $\frac{2s\beta}{\eta^2}$ . . . . .	104
95	ECMS Power Limit Calculations - $P_{lim}^{bo}$ and $P_{lim}^{th}$ . . . . .	104
96	ECMS Power Limit Calculations - $P_{lim}^{re}$ . . . . .	105

## SUMMARY

Governments around the world are slated to pass legislations that progressively decrease automobile  $CO_2$  and  $NO_x$  emissions. Automotive manufacturers are thus looking for cost effective means to reduce tailpipe emissions while retaining existing powertrain components and reducing the number of component changes. Hybrid electric vehicles are hence an attractive solution. The presence of at least two power converters gives an additional degree of freedom compared to conventional vehicles. In order to optimize fuel consumption, an appropriate control strategy needs to be developed.

The objective of this thesis is to develop an energy control strategy that addresses the additional degree of freedom, but is also causal and computationally efficient so that it can be implemented on a real vehicle.

First a detailed plant model that best represents the vehicle is developed in Simulink. Then a simple heuristic supervisory controller is developed to understand and get familiar with the various hybrid operating modes.

As a third step, an optimal control strategy using Pontryagin's minimum principle presented in [16] is considered. The author of [16] derives an explicit control law by considering a simplified powertrain model. This results in the identification of power threshold limits that separate the different modes of operation. The optimal power that needs to be delivered by the internal combustion engine and the electric machine are also obtained as equations for each operating mode. Hence the optimization is done at design time and is computationally inexpensive. Evaluation of this strategy

shows that it achieves good performance in terms of fuel consumption. The fuel consumption differs slightly when considering driving cycles with elevation profiles.

To achieve better fuel consumption in environments with changing elevation profiles, the control strategy is extended to consider some knowledge about future driving conditions. An algorithm that generates a reference trajectory for the future state-of-charge of the battery considering the future elevation and speed profiles is used along with the previously proposed strategy [16]. The average speeds and elevation change data for a planned trip are provided by a navigation device. The predictive strategy is computationally efficient and the resulting fuel consumption are improved compared to the non-predictive strategy.

# CHAPTER I

## INTRODUCTION

Legislations around the world are tending towards low tailpipe emission limits and hence hybrid electric vehicles are an attractive choice for automakers without having to make significant changes to existing powertrain components in general. The fuels used in conventional vehicles provide the advantage of high energy density but suffer from poor operating efficiency. Hybridization of the powertrain helps reduce fuel consumption in many ways. The presence of an electric machine (EM) allows for reduction in the size of internal combustion engines (ICEs). During deceleration, energy normally lost in the form of heat from brakes can be recuperated by operating the EM (EM) in generation mode. The ICE can be shut off during times of coasting to avoid overrun and idling losses. The operating point of the ICE can be maintained in high efficiency regions during part-load conditions by further loading the engine with the EM operated in generation mode. During low-load conditions the ICE can be shut off and the vehicle can be operated in a pure electric mode.

The key to achieving the best possible efficiency is to set the most optimal power split ratio between the ICE and the EM to meet the driver demand. The algorithms used to determine this ratio are called energy management strategies [1]. These strategies can be heuristic or optimal in nature.

Heuristic approaches are rule-based approaches that are very sensitive to calibration parameters [2, 3, 4]. The calibration used to reduce fuel consumption for standard drive cycles need not always work for real-time driving conditions. Hence there is



always room to improve the fuel efficiency further.

Optimal energy management strategies are based on theories of optimal control and are derived for a model that represents the vehicle. These strategies minimize a local cost function at every operating condition to find an input control signal for the plant. Such a cost function is typically an equivalent fuel consumption that represents the energy consumed by using fuel and electric power. These approaches are introduced by the authors in [5, 6, 7] and improved in [8, 9]. These strategies need a model of the powertrain and are hence scalable, but require a lot of computational power since an optimization of the cost function needs to be carried out in real-time. Pontryagin's minimum principle can be used to derive strategies that minimize a local cost function with lower computational power [10, 11].

Any energy management strategy can be truly fuel optimal only if the driving conditions and the entire drive cycle are accurately known beforehand [12]. The resulting strategies derived from methods such as Dynamic Programming are not causal and cannot be used in real-time applications. An acausal supervisory control that optimizes the fuel consumption of a hybrid electric bus for repeatedly driven routes is presented in [13].

## ***1.1 Motivation***

EcoCAR3 is a collegiate automotive engineering competition organized by the US Department of Energy, that challenges 16 North American university teams to redesign a Chevrolet Camaro as a hybrid electric vehicle to reduce its environmental impact, while maintaining the performance of the car. The students are given the stock Camaro by GM. They need to retrofit the vehicle with hybrid components and develop a control strategy to meet performance and emission criteria. This thesis is a

result of the development of a control strategy for the parallel hybrid vehicle designed by the team at Georgia Institute of Technology.

The control strategy presented in this thesis only considers the optimization of the power delivered by the ICE and EM for a given driver demand. The gear shift strategy described in Appendix A is used for all the energy management strategies presented. The control strategies presented were only evaluated in a simulation environment and have not been implemented in real-time.

## ***1.2 Structure of the Thesis***

In Chapter 2, a representation of the hybrid vehicle is modeled in Simulink. To simulate the vehicle in real-world driving conditions, real-world routes are selected and the corresponding road gradient profiles are extracted in Chapter 3. To understand the operation and the various operating modes of the vehicle, a simple supervisory controller based on heuristics is developed in Chapter 4. An optimal control strategy developed using Pontryagin's minimum principle is studied in Chapter 5. This strategy is further extended in Chapter 6 to take into account future driving conditions such as road elevation profiles and expected average traveling speed obtained from a navigation device. The main findings of the thesis are summarized in Chapter 7. The optimal and predictive control strategies presented in this thesis are based on [14, 15, 16].

## CHAPTER II

### VEHICLE MODEL FOR A PARALLEL HEV

The vehicle modeled in this thesis is a 2016 Chevrolet Camaro that was converted to a P3 parallel hybrid electric vehicle (HEV) by the EcoCAR3 team at Georgia Institute of Technology. If the EM is coupled to the drivetrain between the gearbox and the differential, it is called a P3 hybrid. In a parallel hybrid vehicle, the ICE and the EM both provide torque to the wheels. The stock 3.6 liter Camaro engine was replaced by a 2 liter engine. The ICE is coupled to the Chevrolet 8-speed automatic transmission (8L45) through a torque converter. The transmission is in turn coupled by means of a prop shaft to a differential that drives the rear wheels. The EM (Parker-GVM 210-150P) is coupled to the prop shaft through a Borg Warner transfer case.

The ICE can be decoupled from the wheels by remotely shifting the transmission to neutral through a command from the supervisory controller. This allows the vehicle to be operated in a pure electric mode where the wheels are powered by just the EM without dragging the ICE and the transmission. The vehicle architecture is illustrated in Figure 1.

The vehicle is modeled in Simulink [17]. Effects caused by changes in temperatures of the battery, the EM, and the ICE are neglected throughout this work. The components shown in Figure 1 are discussed in the following sections.

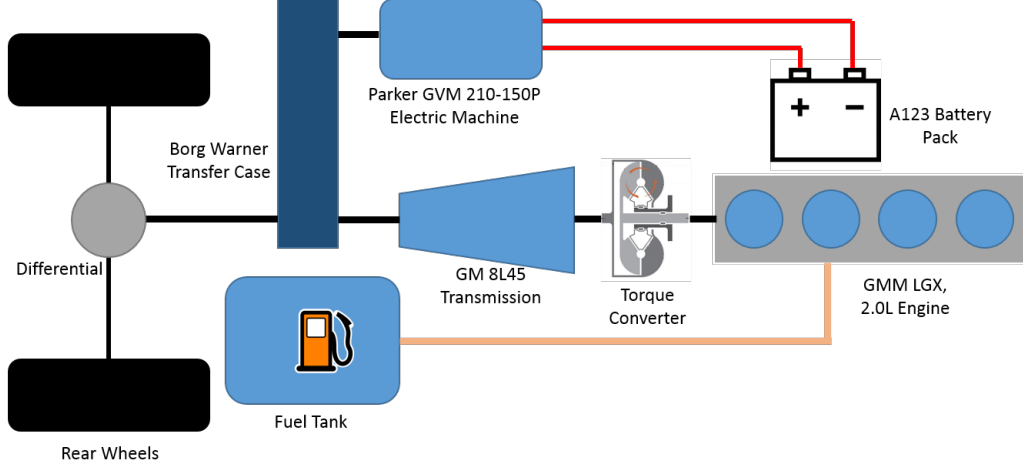


Figure 1: GT EcoCAR3 Vehicle Architecture

## 2.1 Chassis Model

The vehicle body, tires, brakes and differential are modeled using configurable blocks from the Simulink Driveline library.

The vehicle body block models a two-axle vehicle body in longitudinal motion. The block accounts for body mass, aerodynamic drag, road incline, and weight distribution between axles due to acceleration and road profile. The block has physical ports for horizontal motion of the vehicle, and also for front and rear normal wheel forces. The model also takes headwind speed and road inclination angle as physical inputs.

The tire block models the longitudinal behavior of a highway tire characterized by the tire Magic Formula. Connection A is the mechanical rotational conserving port for the wheel axle. Connection H is the mechanical translational conserving port for the wheel hub through which the thrust developed by the tire is applied to the vehicle. Connection N is a physical signal input port that applies the normal force acting on the tire. The force is considered positive if it acts downwards. The parameters used to calibrate the tires are given in Table 1.

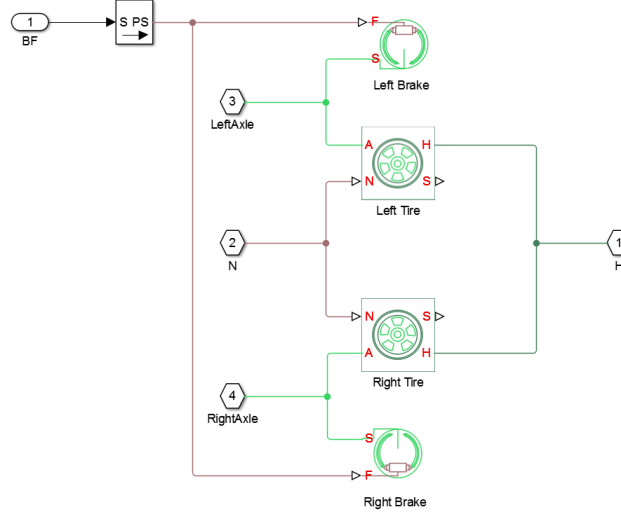


Figure 2: Model for Tires and Brakes

Table 1: Tire Parameters

Parameter	Value	Unit
Tire Radius	0.341	m
Tire Pressure	35	psi
Tire Alpha	-0.473601	-
Tire Beta	1.01360	-
Tire A Coefficient	0.0684	-
Tire B Coefficient	$1.65 \times 10^{-4}$	-
Tire C Coefficient	$4.12 \times 10^{-7}$	-

The 'Double-Shoe Brake' block is used to model the brakes on the wheels. This block models a brake arranged as two pivoted rigid shoes symmetrically installed inside or outside a drum and operated by one actuator. The actuator force causes the shoes to exert a friction torque on a shaft connected to the drum. The block has physical ports representing the conserving rotational terminal associated with the drum shaft and force input to the actuator. Positive force creates friction torque that resists shaft rotation. The Simulink implementation of the tires and their interface with the brakes is shown in Figure 2.

The differential is modeled as a planetary bevel gear train equipped with an additional bevel gear transmission between the driveshaft and the carrier. The pinion gear of

this transmission is attached to the driveshaft while the large bevel crown gear is affixed to the carrier.

The front and rear wheels are connected to the vehicle body block as shown in Figure 3. The rear wheels are connected to the output of the differential with inertia blocks that model the driveshaft inertias.

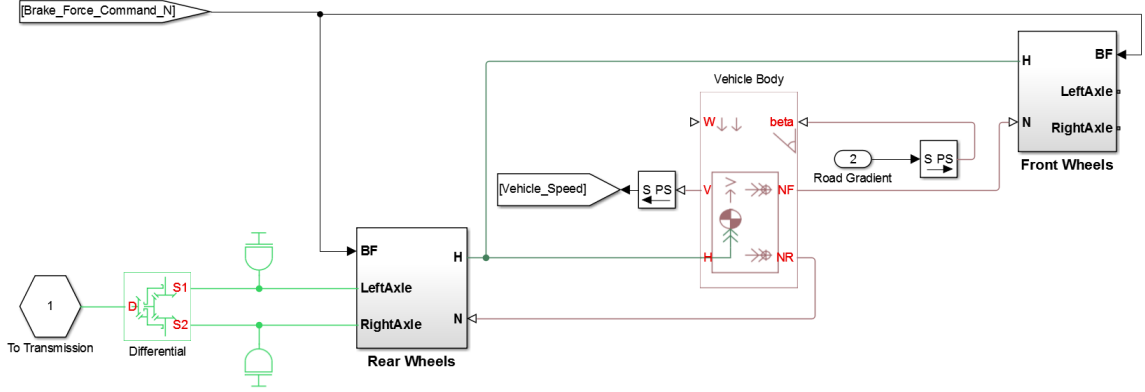


Figure 3: Chassis Model

## 2.2 Transmission and Torque Converter

The 8-speed transmission is modeled as shown in Figure 4. Simple gears are used to model the different gear ratios of the transmission. These gears are engaged or disengaged between the input and output shafts using 4 double-sided synchronizers. Each synchronizer is modeled as two back-to-back dog clutches, two back-to-back cone clutches, and a detent. As the shift linkage translates in the negative direction, the clutches engage the ring with hub A. This engages the odd gear. When the linkage translates in the positive direction, the clutches engage the ring with hub B. This engages the even gear. When the magnitude of the shift linkage translation is less than the cone clutch ring-hub gap, the synchronizer is in neutral and does not transmit torque.

Table 2: Gearbox Ratios

Parameter	Value
Gear 1	4.62
Gear 2	3.04
Gear 3	2.07
Gear 4	1.66
Gear 5	1.26
Gear 6	1.00
Gear 7	0.85
Gear 8	0.66
Final Drive Ratio	3.27

Table 3: Gearbox Inertias

Parameter	Value	Unit
Engine Clutch Inertia	0.08	$\text{kgm}^2$
Transmission Clutch Inertia	0.01	$\text{kgm}^2$
Input Shaft Inertia	0.028	$\text{kgm}^2$
Output Shaft Inertia	0.01	$\text{kgm}^2$

The synchronizer linkages are usually powered by hydraulics, but here they are represented by a simplified model as shown in Figure 4. The shift linkages are operated based on gear selection control input received from the transmission control module (TCM). A description of the strategy used in the TCM to determine transmission gear position is given in Appendix A.

The input and output shaft inertias are modeled using inertia blocks. The torque converter is modeled as a three-part torque converter consisting of impeller, turbine, and stator. The block was configured using the speed ratio, torque ratio and capacity factor data provided by GM. The torque converter is connected between the transmission and ICE. The gear ratios of the gearbox are given in Table 2. The inertias considered while modeling the gearbox are given in Table 3.

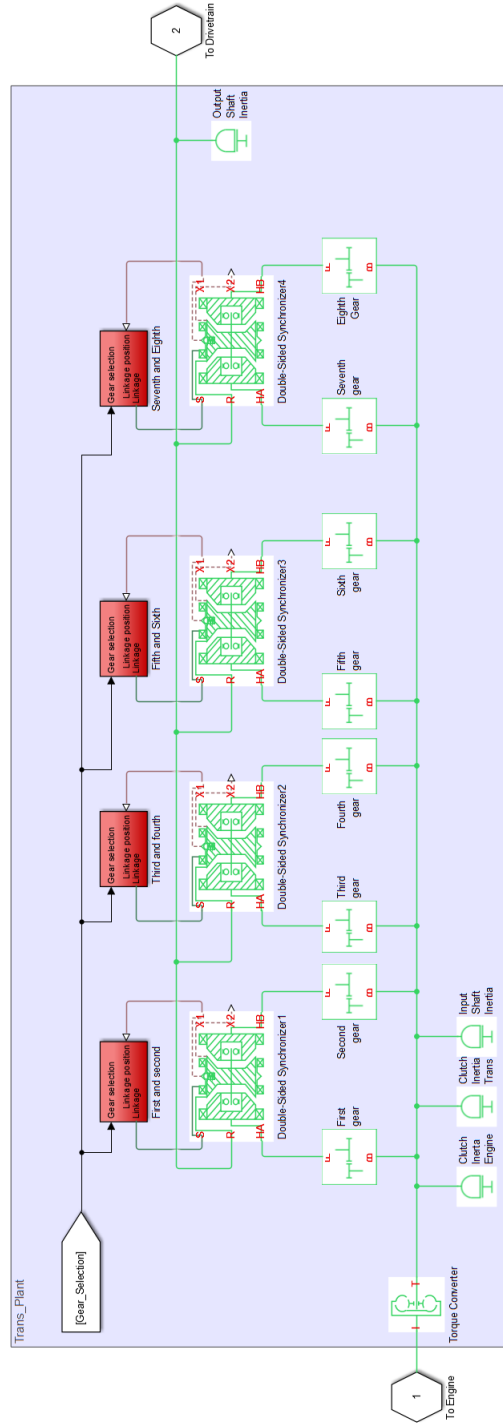


Figure 4: Transmission Model

### 2.3 Internal Combustion Engine

The ICE block models the core engine along with the throttle body and the air intake manifold.



The air flow through the throttle body is modeled as the flow rate of an ideal gas through a sharp-edged variable-area orifice as shown in Figure 5. The throttle delay is modeled using a first-order filter whose input is the throttle position command from the supervisory control. The orifice area is obtained using a look-up table that uses the output of the first-order filter as the input. A constant volume pneumatic chamber is used to model the air intake manifold. This is shown in Figure 6.

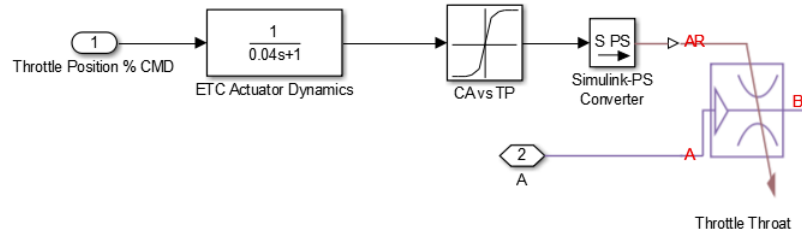


Figure 5: Throttle Body

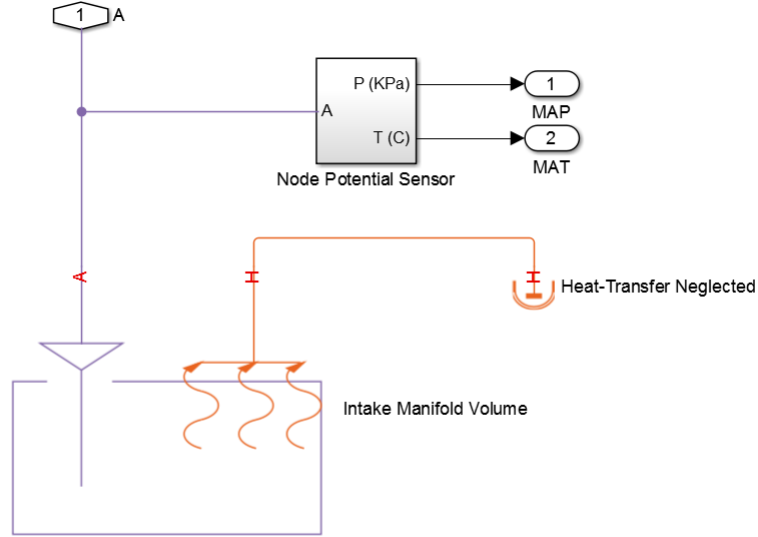


Figure 6: Air Intake Manifold

The ICE volumetric efficiency is determined from a map that uses ICE speed and manifold air pressure as inputs. The engine intake port flow is computed from the volumetric efficiency using the ideal gas equation. The maps are calibrated using data provided by GM. The physical air flow through the engine is modeled using

a controlled pneumatic flow rate source connected to the throttle body and intake manifold. The block uses the calculated intake port flow value as the input. The core engine model is shown in Figure 7.

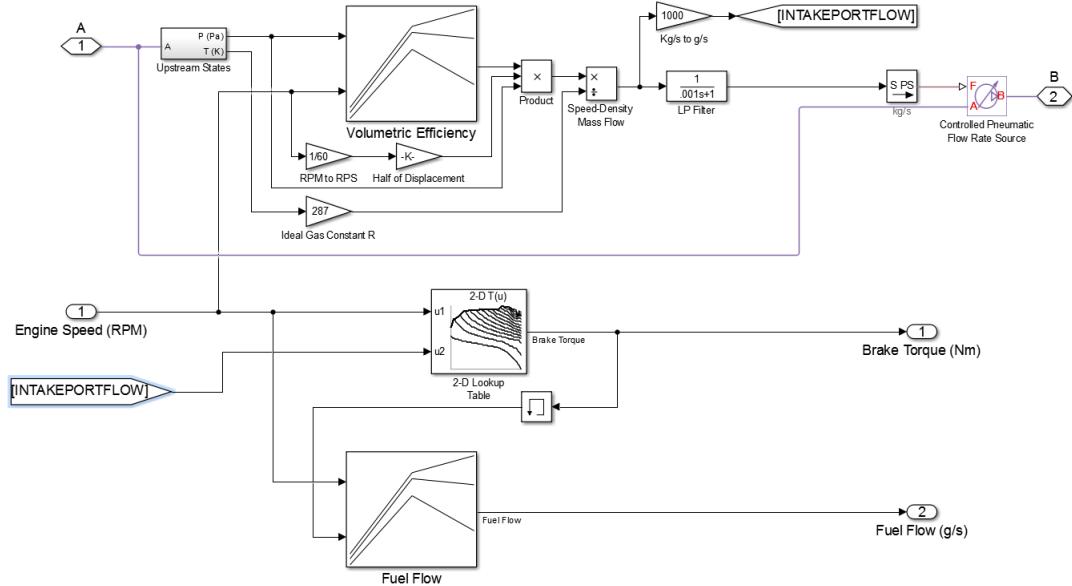


Figure 7: Core Engine Model

The brake torque developed by the engine at the crankshaft is calculated as a function of the engine speed and intake port flow using a map. The physical torque developed by the engine is modeled using an ideal torque source that uses this value as an input. The fuel flow rate is calculated using a map which is a function of engine speed and crankshaft torque. The starter motor is modeled simply as an ideal torque source and is connected to the engine crankshaft. This is shown in Figure 8. The starter motor torque request and the engine throttle request used by the throttle body are set by the engine control module as explained in Appendix B.

The exhaust system is modeled using a constant area pneumatic orifice. The engine inertia is accounted for by an inertia block connected to the crankshaft output. The brake torque and power at engine crankshaft are shown in Figure 9.

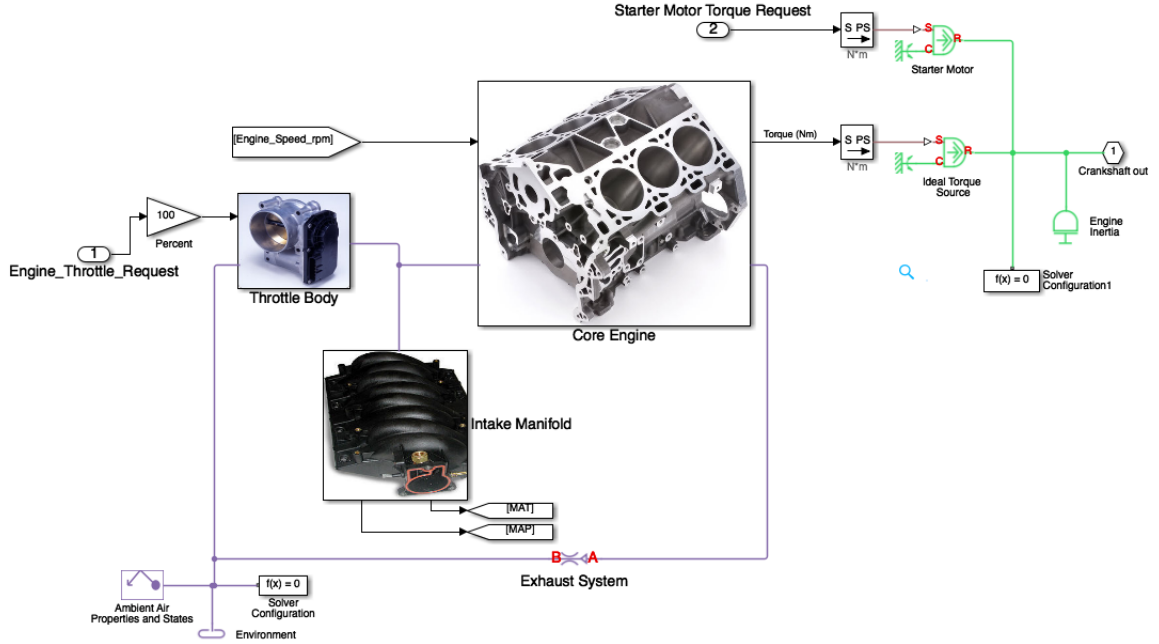


Figure 8: Overall Engine Model

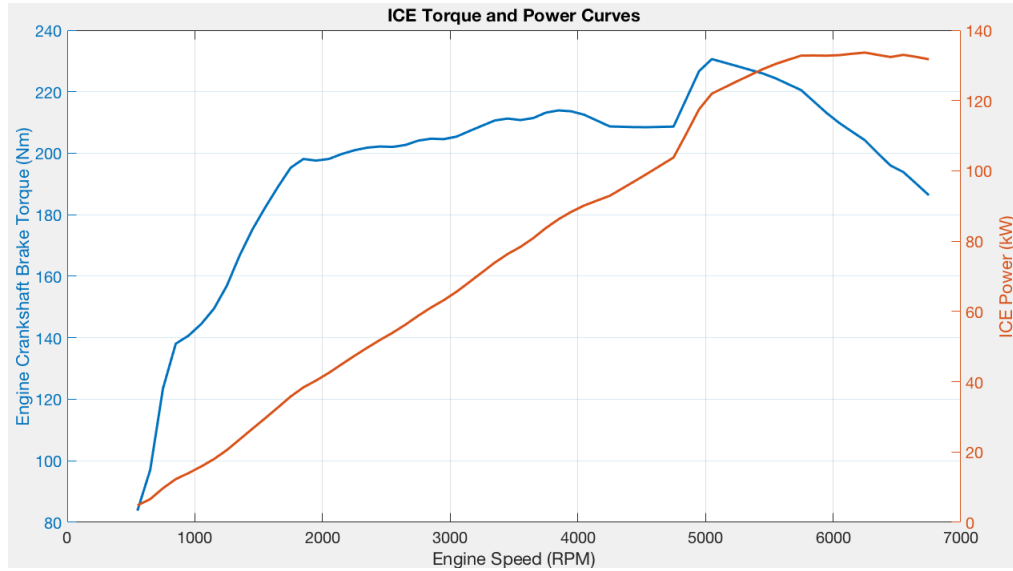


Figure 9: Engine Torque and Power

## 2.4 Electric Machine

The EM used is a Parker GVM 210-150P three phase permanent magnet synchronous machine (PMSM). In reality the EM is driven by a three phase inverter connected to

a DC source, which is the battery pack on the vehicle. Modeling the three phase operation of the EM and the inverter would introduce complexity and lengthen simulation time as the semiconductor switching occurs at time steps in the order of microseconds. Modeling these dynamics does not greatly affect the energy consumption of the vehicle. Hence, without loss of generality, the currents, torque and power drawn by the EM are calculated by simple mathematical relations. The inverter efficiency is approximated to 0.9.

The peak torque the EM can generate is determined from a look-up table that uses the EM speed as the input. The table is calibrated using data from the component data sheet. This torque is multiplied with a normalized torque command request determined by the supervisory controller to calculate the operating torque. The physical torque generated by the EM is modeled by an ideal torque source that uses the calculated operating torque as the input. The inertia of the EM is modeled using an inertia element connected to the output of the torque source. This is shown in Figure 10. The EM torque vs speed curve is shown in Figure 11. The operating efficiency of the EM is calculated, as shown in Figure 12, by two maps depending on whether the EM is in motoring or generating mode. Both maps use EM torque and speed as input and are calibrated from the machine data sheet.

The mechanical power developed by the EM is calculated as the product of the EM torque and speed. This value is then divided or multiplied by the EM efficiency depending on the EM operating mode, to calculate the electric power developed by the EM. The result is multiplied with the inverter efficiency to determine electrical power drawn or delivered to the battery pack. This power is divided by the battery voltage to determine the current drawn by the EM. These calculations are shown in Figure 13.

The effects of temperature and heat exchange are neglected throughout this thesis.

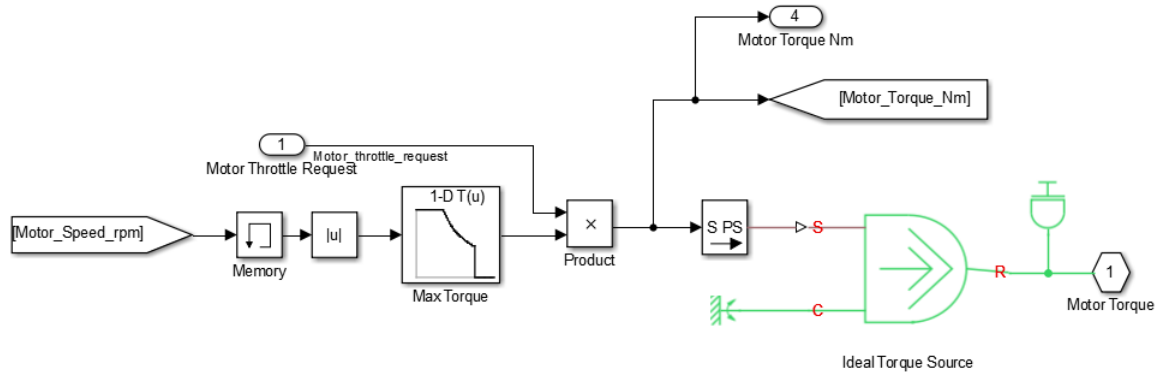


Figure 10: Calculation of EM Torque

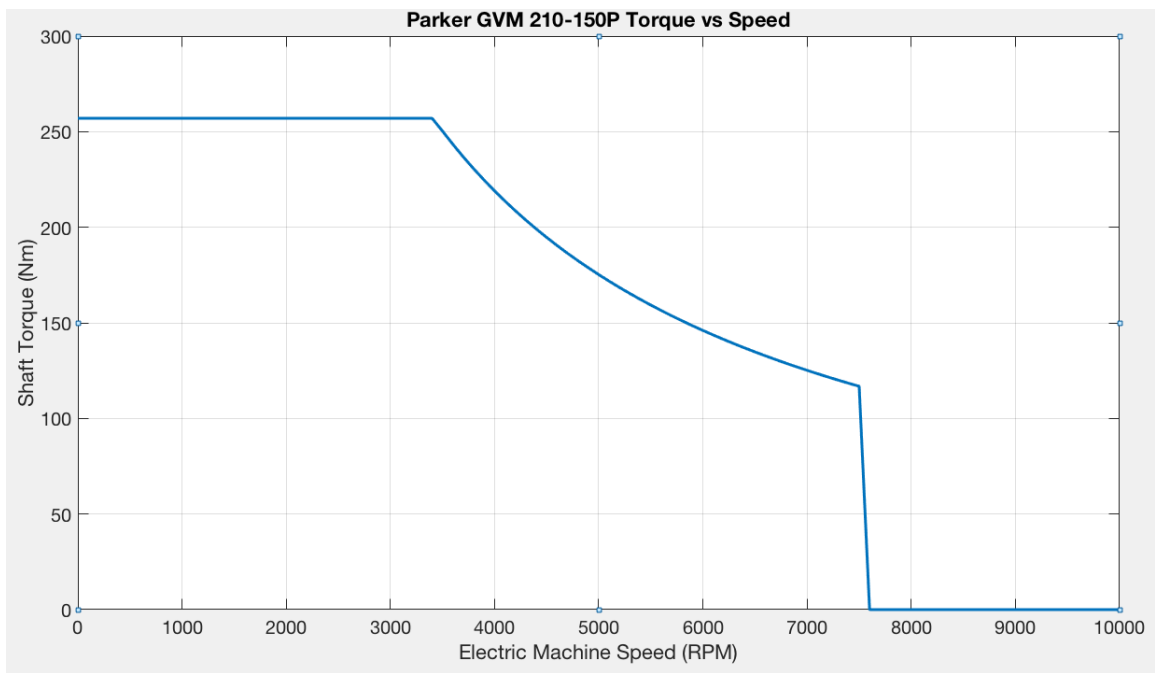


Figure 11: EM Torque vs Speed

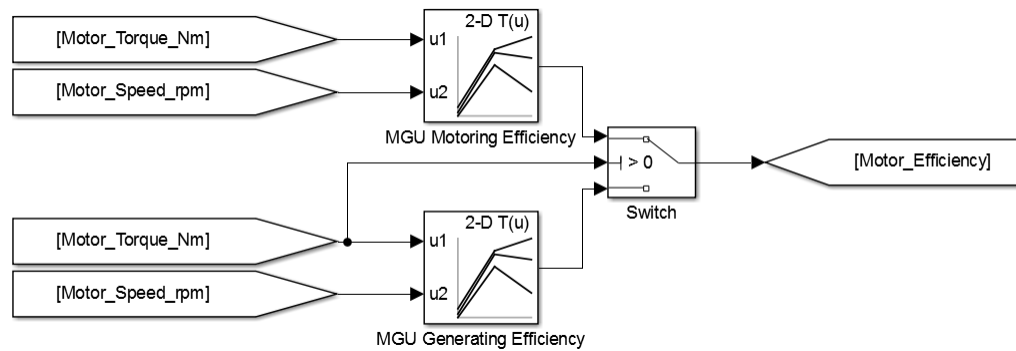


Figure 12: Calculation of EM Efficiency

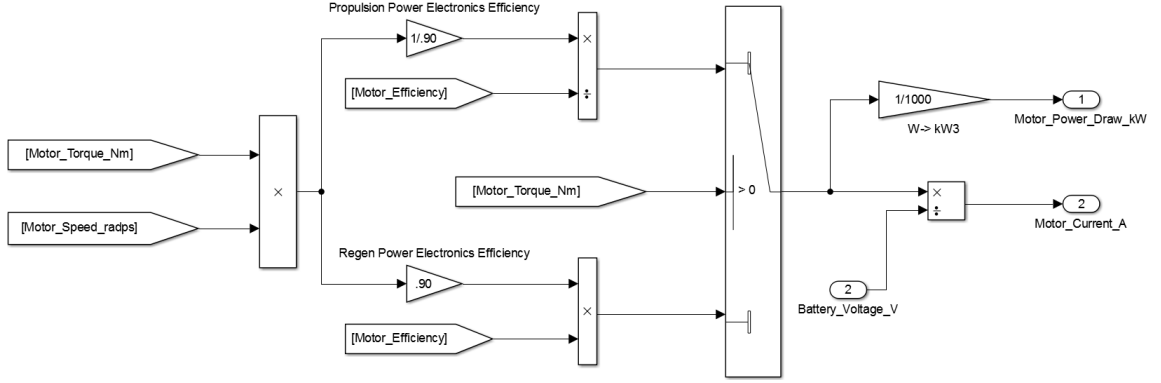


Figure 13: Calculation of EM Power and Current

## 2.5 High Voltage Battery Pack

The battery pack used in the vehicle was built by connecting 7 individual modules from A123. Each module has a 15s2p configuration meaning that it has 2 strings of cells connected in parallel and each string has 15 cells connected in series. The open circuit voltage of the pack is approximately 324V at 300K and 50% state of charge (SOC). The pack capacity is approximately 19.6 Ah<sup>1</sup>.

The charge available in the battery is calculated by integrating the current drawn by the EM over time and adding this to the initial battery charge. The battery SOC is the ratio between the available charge and the pack capacity. This calculation is shown in Figure 14.

The open circuit voltage (OCV) for a single cell is determined from a map that uses battery temperature and battery SOC as inputs. The map is calibrated with data from the component datasheet provided by A123. The cell OCV is multiplied by the number of cells in series to calculate the pack OCV. This is shown in Figure 15.

The internal resistance of the battery is determined from two maps depending on

<sup>1</sup>The distance traveled in the considered drive cycles is a maximum of 16 km. In this thesis the pack capacity is assumed to be 8 Ah so that changes in SOC over a given drive cycle are appreciable in the charge sustain mode.

whether the battery is charging or discharging. Both maps use battery SOC and battery temperature as inputs and are calibrated with data from the component datasheet from A123. This is shown in Figure 16.

The product of current per parallel path and the battery internal resistance gives the battery internal voltage drop. Depending on whether the battery is charging or discharging, the voltage drop is added or subtracted from the battery OCV to calculate the battery terminal voltage. This is shown in Figure 17.

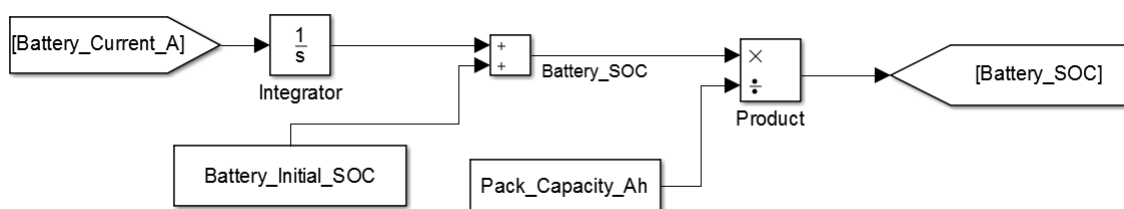


Figure 14: Calculation of Battery SOC

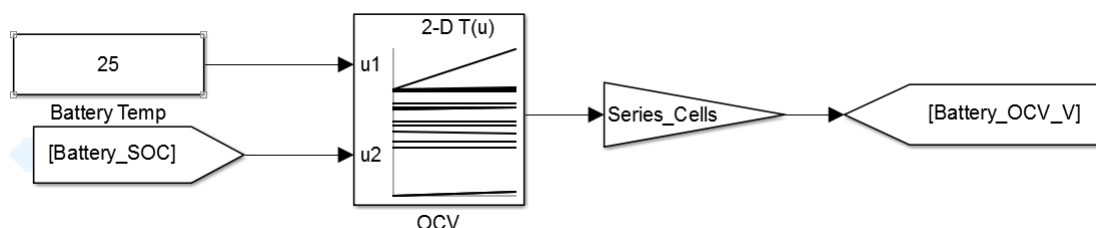


Figure 15: Calculation of Battery OCV

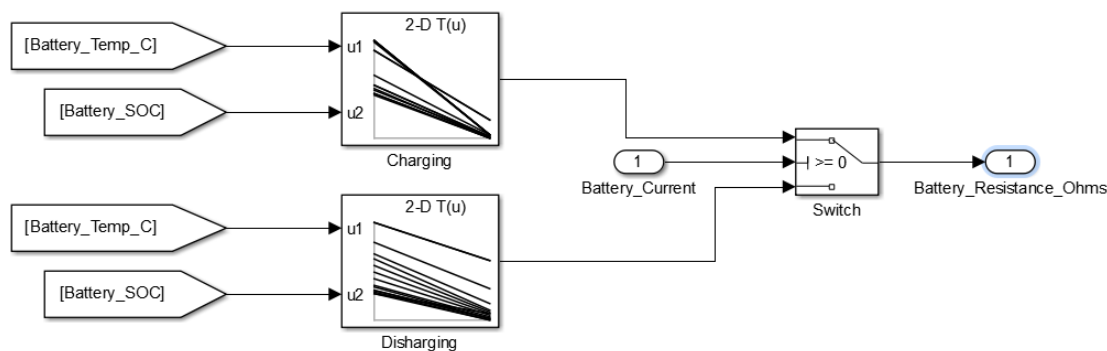


Figure 16: Calculation of Battery Internal Resistance

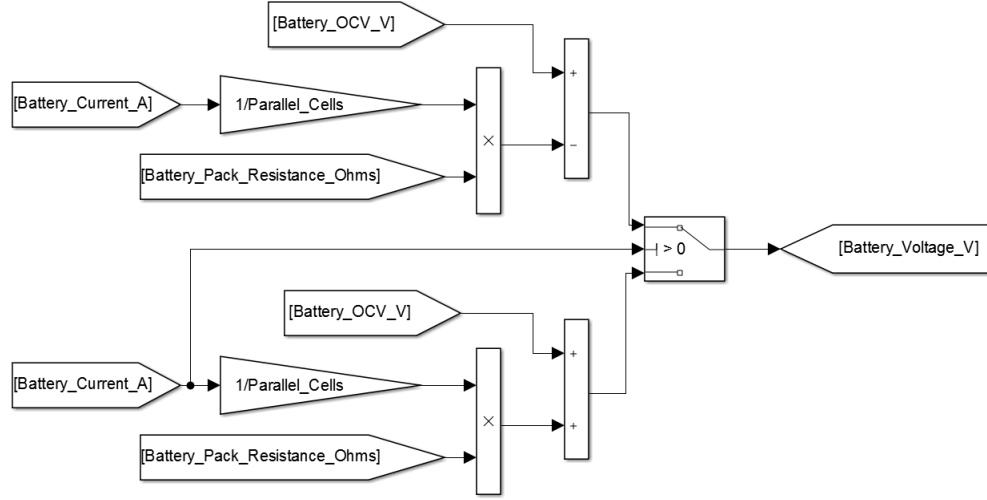


Figure 17: Calculation of Battery Voltage

To derive the SOC dynamics, a Thevenin equivalent circuit is used to model the battery, neglecting electrochemical dynamics. This is shown in Figure 18.  $V_{OC}$  is the

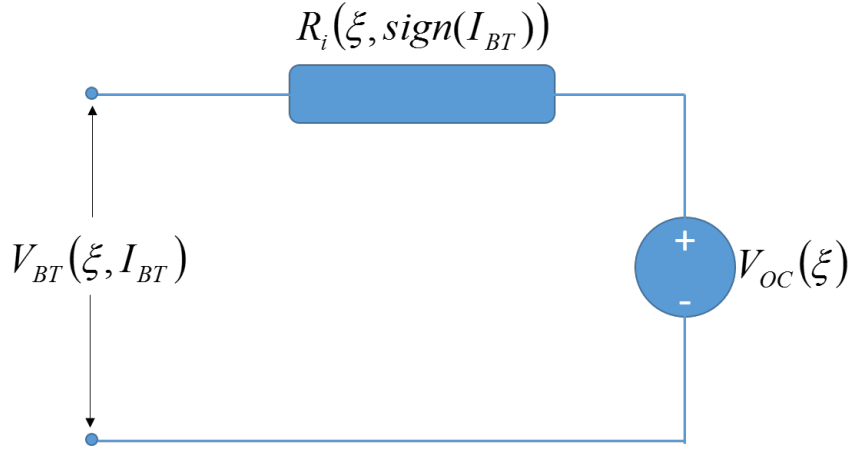


Figure 18: Thevenin's Equivalent Circuit for Battery Model

open circuit voltage of the battery expressed as a function of SOC  $\xi$ . The internal resistance  $R_i$  is expressed as a function of battery SOC and the direction of the battery current  $I_{BT}$ . The terminal voltage of the battery is thus given as:

$$V_{BT}(\xi, I_{BT}) = V_{OC}(\xi) - R_i(\xi, \text{sign}(I_{BT})) \times I_{BT}, \quad (1)$$

where a discharging current is considered positive. Hence the battery power  $P_{BT}$  can



be expressed as:

$$P_{BT} = V_{BT}I_{BT} = V_{OC}I_{BT} - R_i I_{BT}^2 \quad (2)$$

This equation is solved for battery current expressed as a function of power that is drawn from or supplied to the battery yielding:

$$I_{BT} = \frac{V_{OC} - \sqrt{V_{OC}^2 - 4R_i \cdot P_{BT}}}{2R_i} \quad (3)$$

The battery SOC is the only state variable. The state dynamics are given by:

$$\frac{d\xi}{dt} = -\frac{I_{BT}}{Q_0}, \quad (4)$$

where  $Q_0$  is the nominal battery capacity. The initial SOC is set to  $\xi_0$ . The allowable range of the SOC is set to:

$$\xi \in [\xi_{min}, \xi_{max}] \quad (5)$$

Throughout this thesis,  $\xi_{min}$  is set to 0.3 and  $\xi_{max}$  is set to 0.8 to keep the battery SOC in operating regions that ensure battery longevity.

## 2.6 Overall Vehicle Plant Model Overview

The vehicle subsystems described in the previous sections are connected to represent the full vehicle model. The crankshaft output from the ICE is connected to the torque converter input in the transmission and the transmission output is connected to the differential input. The EM shaft is interfaced to the connection between the transmission and differential with a simple gear that represents the transfer case. The connected vehicle plant model is shown in Figure 19.

The inputs to the vehicle plant are the ICE throttle request, EM torque request and the brake force command. These signals are generated by the supervisory controller. The plant model generates outputs such as the ICE speed, ICE torque, battery SOC, vehicle speed and also speed, torque and efficiency of the EM. These signals, along

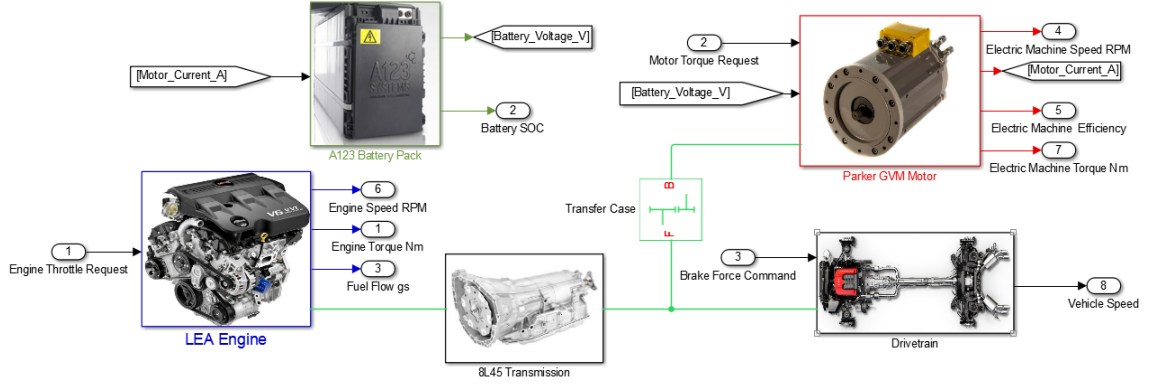


Figure 19: Overall Vehicle Plant Model

Table 4: Vehicle Parameters

Parameter	Value	unit
Vehicle Mass	1809	kg
Frontal Area	0.22	m <sup>2</sup>
Prop Shaft Inertia	0.0015	kgm <sup>2</sup>
Driveshaft Inertia	0.0012	kgm <sup>2</sup>
Transfer Case Ratio	1.686	-
Motor Shaft Inertia	0.03086	kgm <sup>2</sup>
Vehicle Drag Coefficient	0.344	-

with the driver torque request are used by the supervisory control to determine the ICE torque, EM torque and brake force commands.

The parameters used to calibrate the vehicle model are given in Table 4.

## 2.7 Drive Cycle and Driver Torque Request

Different countries and organizations use standardized driving cycles, which are basically vehicle speed as a function of time, to assess vehicles on performance metrics such as fuel consumption and tailpipe emissions. For actual vehicles these tests are carried out on chassis dynamometers where the test driver tries to follow the drive cycle speed within an allowed error limit of 2 miles per hour.

In this thesis, the US06 urban, US06 highway, highway fuel economy driving schedule

(HWFET) and the US505 drive cycles are used to assess the performance of the vehicle that is modeled. The plots of these drive cycles are shown in Appendix C.

The 'Drive Cycle Select' block is configured to allow the user to choose the drive cycle for the simulation. During run time, the block provides the target vehicle speed.

In reality the driver would use the accelerator pedal or the brake pedal to follow the drive cycle. This behavior is modeled in Simulink using a PID controller. The error input to the PID controller is the difference between target vehicle speed and the actual vehicle speed. The output of the PID controller is viewed as a normalized driver torque request. The value is limited between 1 and -6 where a positive value represents a request for accelerating torque and a negative value represents a request for braking. The braking limit is set at -6 to account for the fact that the maximum braking torque of a vehicle is much more than its maximum acceleration torque.

The modeling of drive cycles and driver torque request is shown in Figure 20.

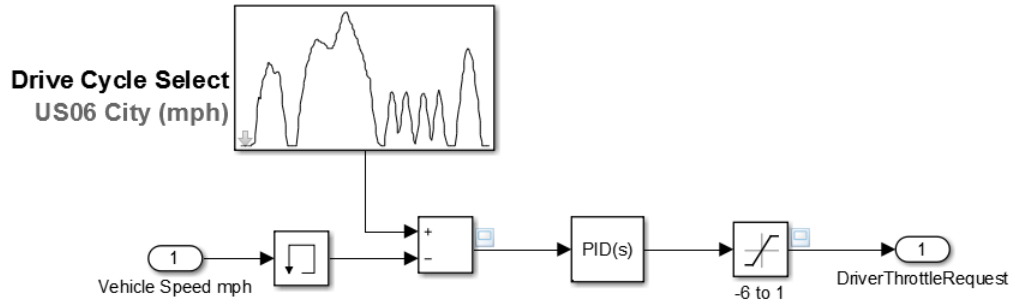


Figure 20: Drive Cycle and Driver Torque Request

## CHAPTER III

### ROUTE SELECTION AND DERIVATION OF ROAD GRADIENT

The vehicle plant model requires the road gradient as an input. To analyze the performance of the supervisory control algorithms presented in this thesis, real-world routes were chosen to consider changes in road gradient.

#### ***3.1 Route Selection***

The following four routes were picked:

- **Route 1:** Zamenhofstrasse 42 to Wernerstrasse 51, Stuttgart Germany.
- **Route 2:** University of Washington to Space Needle in Seattle, USA.
- **Route 3:** Gatlinburg Hills, USA.
- **Route 4:** Route in Sausalito, California, USA.

The screenshots of the routes from Google Maps are shown in Figures 21 to 24.

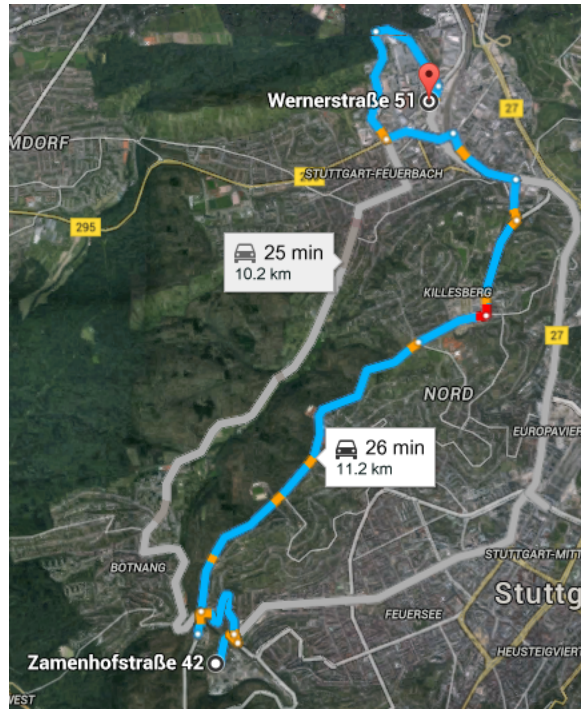


Figure 21: Route from Zamenhofstrasse to Wernerstrasse

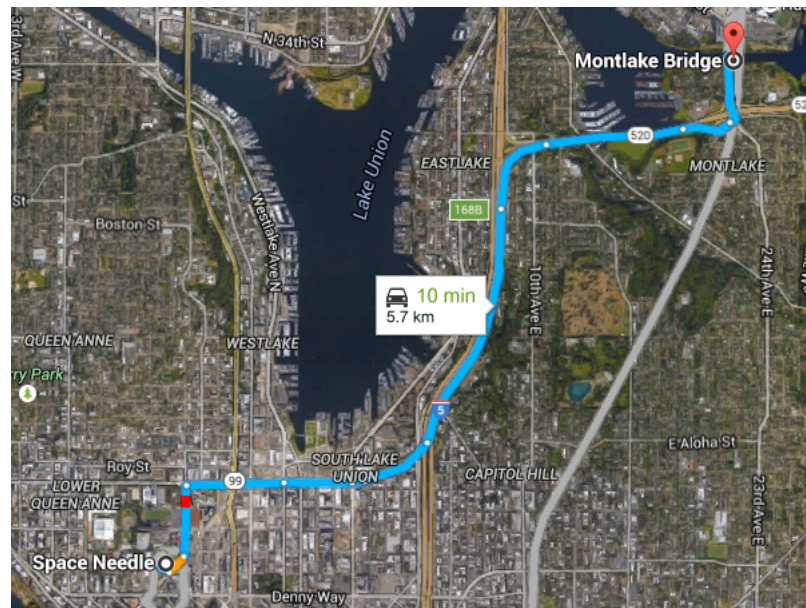


Figure 22: Route from UW to Space Needle in Seattle

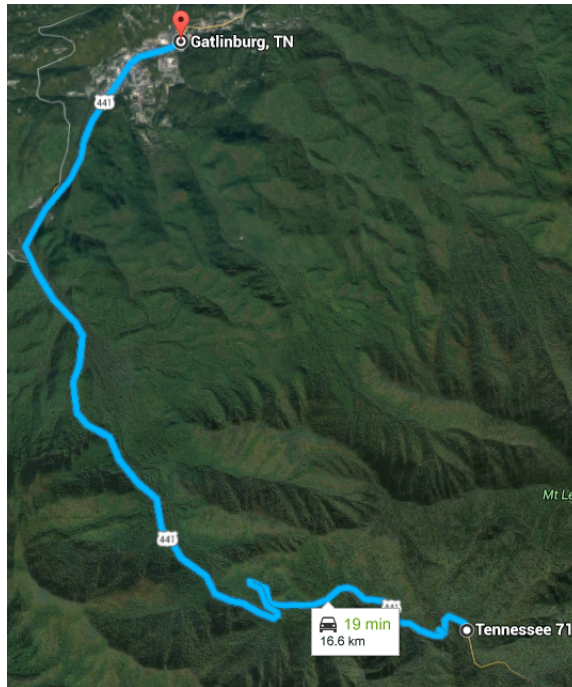


Figure 23: Route in Gatlinburg

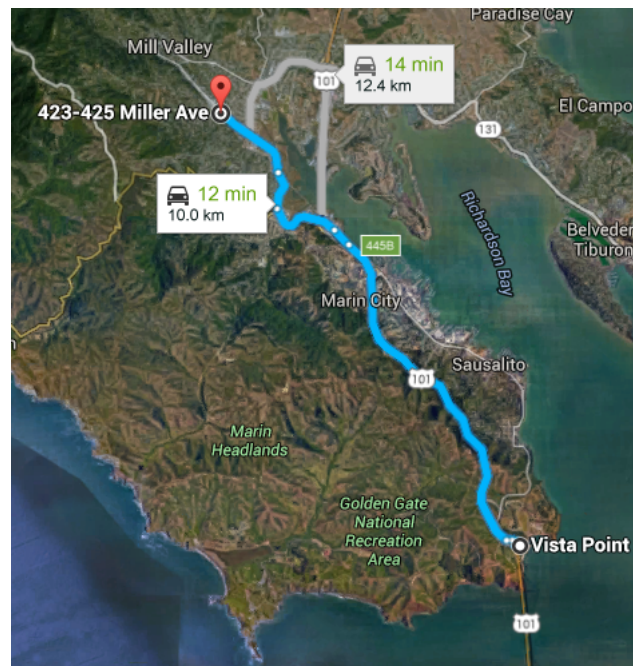


Figure 24: Route in Sausalito California

### 3.2 *Extraction of Road Gradient*

Once the routes are chosen, the road gradient information needs to be extracted. The steps taken to do this are detailed below:

- The route chosen on Google Maps is saved as a KML file. KML (Keyhole Markup Language) is an XML based file format used to display geographic data in an Earth browser such as Google Earth and Google Maps. The KML file provides GPS coordinates in latitude and longitude for the entire route.
- A Matlab script file is written using a string search approach to extract numbers enclosed between the tags '<coordantes>' and '</coordantes>'. These numbers are the latitude and longitude for a point on the route. The latitude and longitude values for all the points that represent the route are stored as arrays.
- Google has developed a service called Google Maps API that provides elevation data for all locations on the surface of the earth, including depth locations on the ocean floor (which return negative values) [18]. An API key needs to be obtained to use the service. The free version of the service allows 2,500 free requests per day, 512 locations per request and 10 requests per second. The elevation at a particular point, for example the location of the GT EcoCAR3 workshop with coordinates (33.787489, -84.406474), can be obtained by typing the string

```
https://maps.googleapis.com/maps/api/elevation/xml?locations=
33.787489,-84.4\06474\&key=YOUR\_API\_KEY
```

in the address bar of a web browser. The string 'YOUR\_API\_KEY' is replaced



with a valid API key. The result returned by the browser is shown in Figure 25. The elevation at this location is 276.7092590 meters above sea level. To obtain

```
▼<ElevationResponse>
  <status>OK</status>
  ▼<result>
    ▼<location>
      <lat>33.7874890</lat>
      <lng>-84.4064740</lng>
    </location>
    <elevation>276.7092590</elevation>
    <resolution>4.7719760</resolution>
  </result>
</ElevationResponse>
```

Figure 25: Result for Elevation Request to Google Maps Elevation API

the elevation at each point in the route chosen, a Matlab script was written to ping the API with the coordinates extracted. The elevation values returned were stored as an array.

- The elevation data is needed as a function of distance traveled from the starting point. Hence, the latitude and longitude coordinates need to be converted to distances. This is done by using the Haversine formula [19] which gives great-circle distances between two points on a sphere from their longitudes and latitudes. For any two points on a sphere, the haversine of the central angle between the points is given by:

$$\text{hav}\left(\frac{d}{r}\right) = \text{hav}(\phi_2 - \phi_1) + \cos(\phi_1)\cos(\phi_2)\text{hav}(\lambda_2 - \lambda_1) \quad (6)$$

where the haversine function *hav* is given by:

$$\text{hav}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{2} \quad (7)$$

In the above equations,  $d$  is the distance between the two points,  $r$  is the radius of the sphere.  $\phi_1$  and  $\phi_2$  are the latitudes in radians of points 1 and 2 respectively.  $\lambda_1$  and  $\lambda_2$  are the longitudes in radians of points 1 and 2 respectively.



The solution to  $d$  is obtained using the inverse haversine or arcsine function:

$$d = r \cdot hav^{-1}(h) = 2r \cdot arcsin(\sqrt{h}) \quad (8)$$

where  $h$  is  $hav(d/r)$ . More explicitly,

$$d = 2r \cdot arcsin\left(\sqrt{hav(\phi_2 - \phi_1) + \cos(\phi_1)\cos(\phi_2)hav(\lambda_2 - \lambda_1)}\right) \quad (9)$$

or

$$d = 2r \cdot arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \quad (10)$$

The elevation profiles obtained for the four selected routes are shown in Figures 26 to 29.

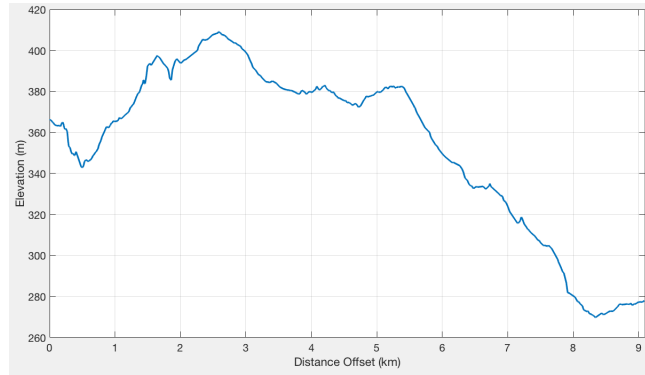


Figure 26: Elevation for Stuttgart Route

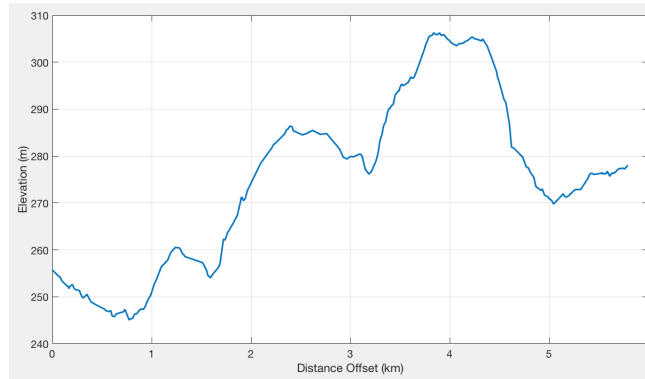


Figure 27: Elevation for Seattle Route

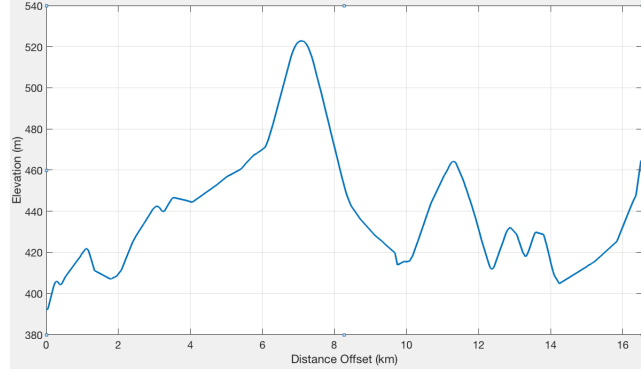


Figure 28: Elevation for Gatlinburg Route

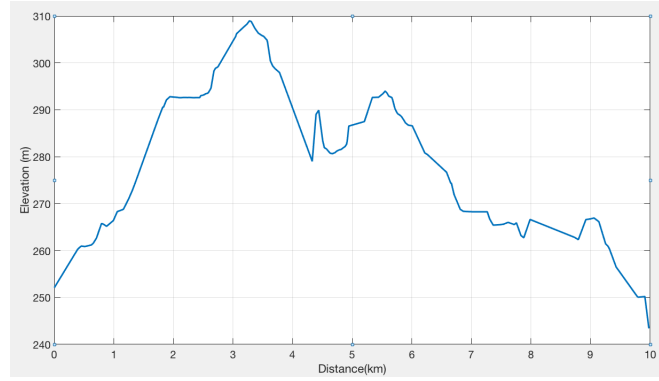


Figure 29: Elevation for Sausalito Route

- Finite differences method is used to calculate the road gradient for each discrete segment using the elevation and distance data:

$$\gamma_i = \tan^{-1} \left( \frac{e_i - e_{i-1}}{d_i - d_{i-1}} \right) \quad (11)$$

where  $\gamma_i$  is the road gradient angle,  $e_i$  is the road elevation at point  $i$  and  $d_i$  is the distance at point  $i$  from the start of the route. The road gradient angle is held constant for that segment.

## CHAPTER IV

### BASIC SUPERVISORY CONTROLLER

The supervisory control uses inputs from the vehicle plant such as ICE speed, EM speed, vehicle speed, battery SOC and gear position, to calculate the ICE torque, EM torque and mechanical braking torque to meet the driver torque request. Many methods exist to perform this calculation. In this section a basic hybrid control strategy based on heuristics is presented.

Plug-in hybrid electric vehicles (PHEVs) typically have two modes of operation based on the battery SOC, namely, charge depletion (CD) and charge sustain (CS) modes. CD mode refers to a mode of vehicle operation that is dependent primarily on energy from the battery pack. The vehicle operates in CD mode at high SOC, and switches to CS mode after the battery has reached a preset lower SOC threshold, exhausting the vehicle's all-electric range. The strategy described in this section uses a blended mode where the ICE supplements the EM during medium and heavy loads.

In CS mode, the vehicle operation is dependent primarily on the ICE and the EM supplements the ICE during heavy loads. In part-load or low-load conditions, when the battery SOC is below the CS set point, the EM is operated in the generating region to recharge the battery. Hence the ICE provides the generating torque in addition to the driver torque request.

## 4.1 Axle Torque Request

The supervisory control receives a normalized driver torque request  $T_{D_{nrm}}$  where a value of 1 represents maximum acceleration and value of -6 represents maximum braking. This is converted to actual axle torque using the maximum ICE and EM torque capabilities.

The maximum ICE torque  $T_{ICE_{max}}$  and maximum EM torque  $T_{EM_{max}}$  are determined using a look-up table each, where the inputs are ICE speed and EM speed respectively. This is shown in Figure 30.

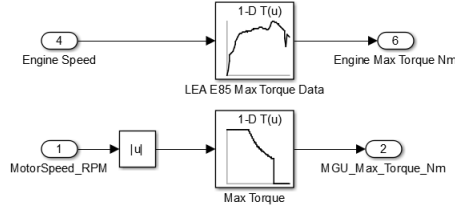


Figure 30: ICE and EM Maximum Torque

The EM torque is further limited based on battery SOC. The maximum EM torque is multiplied by a factor that is linearly faded out from 1 to 0 between 35% and 30% battery SOC. The result is the EM maximum torque limit  $T_{EM_{MaxLim}}$ . This is done to avoid over-depleting the battery.

The maximum generating torque for the EM is the negative of the maximum EM torque as determined from the look-up table. This value is multiplied by a factor that is linearly faded out from 1 to 0 between 75% and 80% battery SOC. The result is the EM minimum torque limit  $T_{EM_{MinLim}}$ . This is done to avoid over-charging the battery.

These calculations are shown in Figure 31. A plot of the fade-off factors is shown in Figure 32.

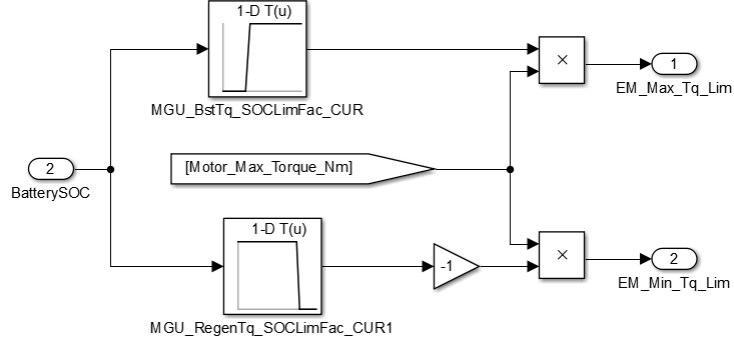


Figure 31: EM Torque Limits

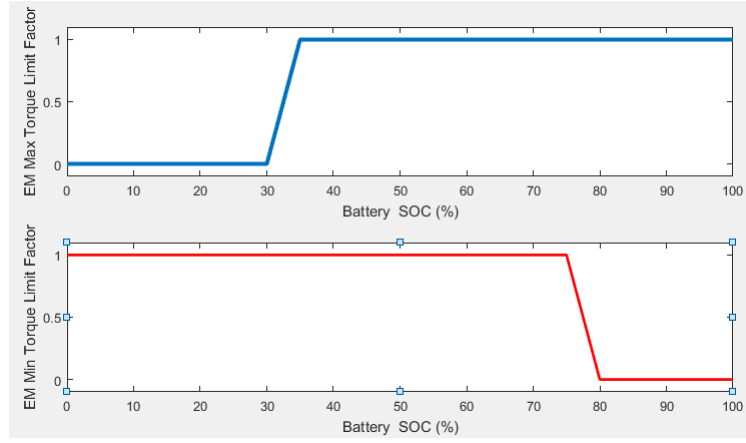


Figure 32: EM Fade Off Factors

The maximum axle torque,  $T_{AxlMax}$ , is given by:

$$T_{AxlMax} = (\zeta T_{ICEMax} + \mu T_{EMMaxLim})\sigma \quad (12)$$

In the above equation,  $\zeta$  is the gear ratio of the current gear,  $\mu$  is the transfer case ratio and  $\sigma$  is the differential ratio.

The axle torque requested,  $T_{AxlReq}$ , is given by:

$$T_{AxlReq} = T_{AxlMax} \times T_{Dnrm} \quad (13)$$

## 4.2 Plant Input

The input  $u$  that the supervisory controller provides to the plant is the portion of the axle torque request that the EM provides:

$$u = \frac{T_{EM_{AxlReq}}}{T_{AxlReq}} \quad (14)$$

This means that the EM component of the axle torque request,  $T_{EM_{AxlReq}}$ , is given by:

$$T_{EM_{AxlReq}} = u \times T_{AxlReq} \quad (15)$$

The ICE component of the axle torque request  $T_{ICE_{AxlReq}}$  is given by:

$$T_{ICE_{AxlReq}} = \begin{cases} (1 - u)T_{AxlReq} & \text{if } T_{AxlReq} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

The brake torque component of the axle torque request,  $T_{Brk_{AxlReq}}$ , is given by:

$$T_{Brk_{AxlReq}} = \begin{cases} (1 - u)T_{AxlReq} & \text{if } T_{AxlReq} < 0 \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

## 4.3 Charge Depletion Mode

The individual contributions of the maximum torque available at the axle at a given wheel speed are calculated using the gear ratio of each gear in the transmission, the transfer case gear ratio and the differential gear ratio along with the torque vs speed characteristics of the ICE and the EM.

The ICE component of the maximum axle torque is given by:

$$T_{ICE_{AxlMax}} = \zeta \sigma T_{ICE_{Max}} \quad (18)$$

The EM component of the maximum axle torque is given by:

$$T_{EM_{AxlMax}} = \mu\sigma T_{EM_{MaxLim}} \quad (19)$$

A plot of ICE and EM components of the maximum axle torque is shown in Figure 33.

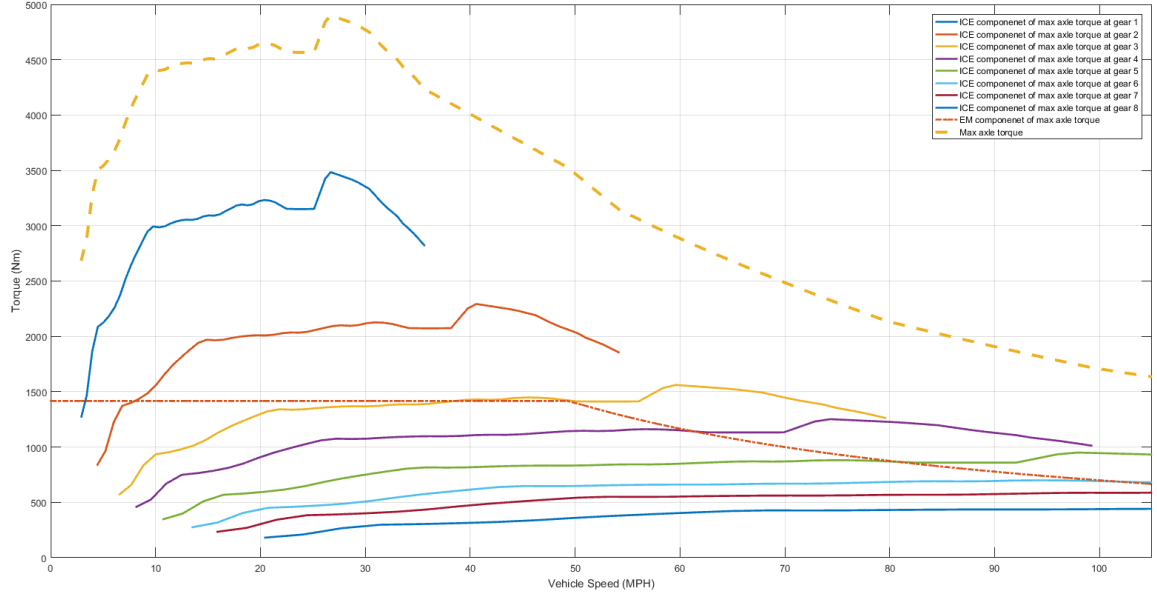


Figure 33: ICE and EM Components of Max Axle Torque

As mentioned earlier, in the CD mode, the battery is the primary source of power and additional power is provided by the ICE during medium and heavy loads. When the driver requests an accelerating torque, i.e., when  $T_{AxlReq} > 0$ , two cases are possible:

- **Case 1:** When  $T_{AxlReq} < T_{EM_{AxlMax}}$ , the torque request is fulfilled by just the EM and the ICE is turned off. This mode is hence called *Electric Mode*. In this mode,  $u$  is set to 1. The ICE torque request,  $T_{ICEReq}$ , is set to 0 and the EM torque request is set to :

$$T_{EMReq} = \frac{T_{AxlReq}}{\mu\sigma} \quad (20)$$

- **Case 2:** When  $T_{AxlReq} > T_{EMAxLim}$ , the ICE needs to provide additional torque to satisfy the driver torque request. This mode is hence called *Dual* mode. The EM torque request,  $T_{EMReq}$ , is set to  $T_{EMMaxLim}$  and  $u$  is given by:

$$u = \frac{\mu\sigma T_{EMMaxLim}}{T_{AxlReq}} \quad (21)$$

The ICE torque request  $T_{ICEReq}$  is set to :

$$T_{ICEReq} = \frac{T_{AxlReq} - \mu\sigma T_{EMMaxLim}}{\zeta\sigma} \quad (22)$$

When the driver requests a braking torque, i.e., when  $T_{AxlReq} < 0$ , again two cases are possible:

- **Case 1:** When  $T_{AxlReq} > T_{EMAxLim}$ , the torque request is fulfilled by regenerative braking and hydraulic brakes are not used. This mode is called *Regeneration*. Hence  $u$  is set to 1, the axle brake torque request  $T_{BrakeReq}$  is set to 0 and EM torque request is given by:

$$T_{EMReq} = \frac{T_{AxlReq}}{\mu\sigma} \quad (23)$$

- **Case 2:** When  $T_{AxlReq} < T_{EMAxLim}$ , the hydraulic brakes are also used. This mode is just called *Braking*. Hence, the EM torque request,  $T_{EMReq}$ , is set to  $T_{EMMinLim}$  and  $u$  is given by:

$$u = \frac{\mu\sigma T_{EMMinLim}}{T_{AxlReq}} \quad (24)$$

The brake torque request is given by:

$$T_{BrakeReq} = -(T_{AxlReq} - \mu\sigma T_{EMMinLim}) \quad (25)$$

In a convetional vehicle, braking is achieved only through the hydraulic brake system which is actuated when the driver presses the brake pedal. Parallel HEVs



also have the capability for regenerative braking. The progressive regenerative braking described above can be implemented independent of the hydraulic brakes by the position of the accelerator pedal instead of the brake pedal. Such an algorithm is called one-pedal-driving algorithm and is described in [26].

The ICE and EM expect normalized torque requests, i.e., a value between 0 and 1 for the ICE and a value between -1 and 1 for the EM. The normalized ICE torque request is given by:

$$T_{ICEReqNrm} = \begin{cases} \frac{T_{AxlReq} - \mu\sigma T_{EMMaxLim}}{\zeta\sigma T_{ICEMaxLim}} & \text{if } T_{AxlReq} > T_{EMAxlMax} \\ 0 & \text{otherwise} \end{cases} \quad (26)$$

The normalized EM torque request is given by:

$$T_{EMReqNrm} = \begin{cases} \frac{T_{AxlReq}}{\mu\sigma T_{EMMaxLim}} & \text{if } T_{EMAxlMax} > T_{AxlReq} > T_{EMAxlMin} \\ 1 & \text{if } T_{AxlReq} > T_{EMAxlMax} \\ -1 & \text{if } T_{AxlReq} < T_{EMAxlMin} \end{cases} \quad (27)$$

In this section, the conditions mentioned for the different cases to be active are purely analytical and do not include hysteresis to avoid any unrealistic frequent switching between modes. This is done in a later section where the actual state machine is developed.

#### 4.4 *Charge Sustain Mode*

In the CS mode, the ICE is the primary source of power and additional power is provided by the EM during medium and heavy loads. When the driver requests an accelerating torque, i.e. when  $T_{AxlReq} > 0$ , three cases are possible:

- **Case 1:** Consider a case where the battery SOC is less than the reference SOC that the controller tries to maintain and the ICE operates in low-load condition (in this case,  $T_{AxlReq} < 0.75 \times T_{ICEMax}$ ). The operating efficiency of the ICE

is poor in low-load conditions. The EM is operated in the generating region to move the ICE close to full-load regions where the ICE efficiency is better. This mode is hence called *Generation*. Here  $u$  is given by:

$$u = 1 - \frac{0.9 \times T_{ICE_{AxlMax}}}{T_{AxlReq}} \quad (28)$$

The EM torque request,  $T_{EMReq}$ , is given by:

$$T_{EMReq} = \frac{T_{AxlReq} - 0.9 \times \zeta \sigma T_{ICE_{AxlMax}}}{\mu \sigma} \quad (29)$$

$T_{ICEReq}$  is set to :

$$T_{ICEReq} = \frac{T_{AxlReq} - \mu \sigma T_{EMReq}}{\zeta \sigma} \quad (30)$$

- **Case 2:** When  $T_{AxlReq} > T_{ICE_{AxlMax}}$ , the EM needs to provide additional torque to satisfy the driver torque request. So this mode is called *Boost*. Hence  $T_{ICEReq}$  is set to  $T_{ICE_{Max}}$  and  $u$  is given by:

$$u = 1 - \frac{\zeta \sigma T_{ICE_{Max}}}{T_{AxlReq}} \quad (31)$$

Consequently,  $T_{EMReq}$  is set to:

$$T_{EMReq} = \frac{T_{AxlReq} - \zeta \sigma T_{ICE_{Max}}}{\mu \sigma} \quad (32)$$

- **Case 3:** When  $0.9 \times T_{ICE_{AxlMax}} > T_{AxlReq} > 0.65 \times T_{ICE_{AxlMax}}$ , the driver torque demand is satisfied just by the ICE. This mode is called *ICE*. Here  $u$  and  $T_{EMReq}$  are set to 0.  $T_{ICEReq}$  is set to:

$$T_{ICEReq} = \frac{T_{AxlReq}}{\zeta \sigma} \quad (33)$$

Braking torque requests from the driver are handled the same way as in CD mode.

In summary, the normalized ICE torque request is given by:

$$T_{ICEReqNrm} = \begin{cases} \frac{T_{AxlReq}}{\zeta\sigma T_{ICEMax}} & \text{if } T_{AxlReq} < T_{ICEAxlMax} \\ 0.9 & \text{if } T_{AxlReq} < 0.65 \times T_{ICEAxlMax} \text{ and } \xi < \xi_{ref} \\ 1 & \text{if } T_{AxlReq} > T_{ICEAxlMax} \\ 0 & \text{otherwise} \end{cases} \quad (34)$$

The normalized EM torque request is given by:

$$T_{EMReqNrm} = \begin{cases} \frac{T_{AxlReq} - \zeta\sigma T_{ICEMax}}{\mu\sigma T_{EMMaxLim}} & \text{if } T_{AxlReq} > T_{ICEAxlMax} \\ \frac{T_{AxlReq} - 0.9 \times \zeta\sigma T_{ICEAxlMax}}{\mu\sigma T_{EMMaxLim}} & \text{if } T_{AxlReq} < 0.65 \times T_{ICEAxlMax} \text{ and } \xi < \xi_{ref} \\ \frac{T_{AxlReq}}{\mu\sigma T_{EMMaxLim}} & \text{if } 0 > T_{AxlReq} > T_{EMAxlMin} \\ -1 & \text{if } T_{AxlReq} < T_{EMAxlMin} \\ 0 & \text{otherwise} \end{cases} \quad (35)$$

In this section too, the conditions mentioned for the different cases to be active are purely analytical and do not include any hysteresis to avoid any unrealistic frequent switching between modes. This is done in the next section where the actual state machine is developed.

#### 4.5 Development of State Machine

A Stateflow chart is used to implement the CD and CS modes. The chart uses the following inputs:

- Driver Torque Request
- Battery SOC
- Minimum EM Axle Torque
- Maximum EM Axle Torque

- Maximum ICE Axle Torque
- Vehicle Speed
- Engine Speed
- SOC threshold for CD mode
- SOC threshold for CS mode
- Reference SOC to maintain in CS mode

The SOC threshold for CD mode is set to 0.6 and that for the CS mode is set to 0.55. The reference SOC that the controller tries to maintain in the CS mode is set to 0.55.

When the supervisory controller is in CD mode, i.e. when the  $SOC > 0.6$ , the controller transitions to CS mode only when the SOC falls down to 0.55. Once the controller is in CS mode, it can go back to CD mode only when the SOC rises again to 0.6. This is an incorporation of hysteresis to prevent the controller from constantly switching between CD and CS modes when a single threshold value is used to switch between the modes.

Apart from all the modes listed in the previous two sections, a ground state called *Coasting* is defined where the ICE and Brake torques are set to 0. The EM meets small values of driver torque requests (50Nm). Hence the supervisory controller has eight main states, namely:

- Coasting
- Electric
- Dual

- ICE
- Boost
- Generation
- Regeneration
- Braking

When operating in *Coasting*, *Electric*, *Regeneration* and *Braking* modes, the gearbox is switched to neutral to prevent dragging the inertia of the gearbox and the ICE. Also, the ICE is switched off in these modes to conserve fuel.

When transitioning from these modes to the other modes that require ICE operation, the ICE needs to be started and the gearbox needs to be engaged again. This takes a finite amount of time and hence transition states are introduced to initiate these actions and to ensure that they are complete before moving to the main states. Only the axle torque component requests of ICE, EM and brake torques are calculated in the chart. The conversion to individual normalized torque requests is done outside the chart. Each state also generates the gearbox and ICE enable flag values as outputs, where '1' corresponds to an enable request and a value '0' corresponds to a disable request.

The main states and the transition states used to implement CD strategy is shown in Figure 34.

When the vehicle is at rest on a level road, the controller is in *Coasting* mode. When the vehicle operates in CD mode and the driver torque request is greater than 0, the controller transitions into the *Electric* mode.

When the driver torque request is greater than the maximum EM axle torque, the

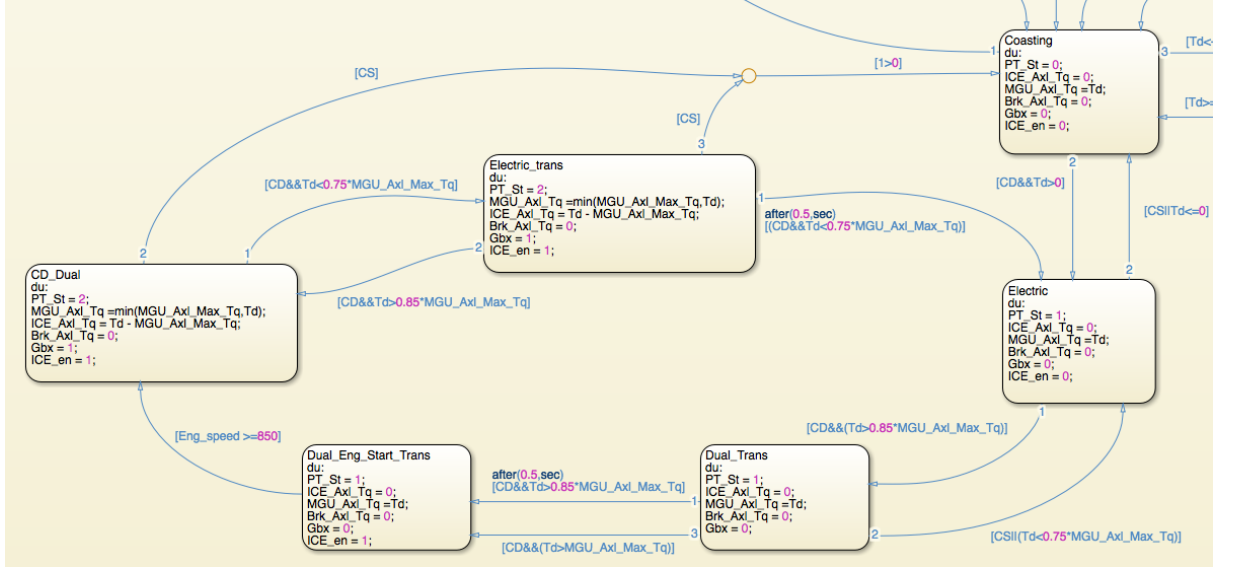


Figure 34: Coasting and CD states

controller needs to be in the *Dual* mode so that the ICE can supplement the EM. As the engine needs to be started and the gearbox needs to be enabled, the controller enters a transition mode called *Dual Transition* when the driver torque request exceeds 85% of the EM maximum axle torque. The values of the outputs calculated in this state are exactly the same as that of the *Electric* mode. If the condition that the driver torque request exceeds 85% of the EM maximum axle torque persists for more than 0.5 seconds, or if the driver torque request exceeds the EM maximum axle torque, the controller enters the *Pre-dual Transition* state. If the driver torque request falls below 75% of the EM maximum axle torque, the controller returns to the *Electric* mode.

The outputs of the *Pre-dual Transition* state are exactly the same as the *Electric* mode except that the ICE enable flag is set to 1. This value is used by the engine control unit (ECM) to start the ICE using the starter motor. When the engine speed exceeds 850 rpm, i.e., when the engine is idling, the controller moves to the *Dual* mode. The gearbox enable flag is set to 1. The controller remains in the *Dual* mode until the driver torque request falls below 75% of the EM maximum axle torque.

Sometimes when the controller is still in the *Dual* state even when the driver torque request is less than the EM maximum axle torque, the ICE will only be idling and the driver torque request is met only by the EM.

When the driver torque request falls below 75% of the EM maximum axle torque, the controller moves from the *Dual* mode to the *Electric Transition* state. The outputs of the *Electric Transition* state are exactly the same as the *Dual* mode. If the condition that the driver torque request falls below 75% of the EM maximum axle torque persists for more than a second, the controller moves to the *Electric* mode where the ICE and the gearbox are disabled. Else if the driver torque request exceeds 85% of the EM maximum axle torque, the control moves back to the *Dual* mode. This is done to reduce the number of engine start and stop events.

When in the *Electric*, *Electric Transition* or *Dual* modes, if the vehicle operation switches to CS mode, the controller moves to the *Coasting* mode from where the CS states can be reached.

The main states and the transition states used to implement CS strategy are shown in Figure 35.

When the vehicle is at rest on a level road, the controller is in *Coasting* mode. When the vehicle operates in CS mode and the driver torque request is greater than 50 Nm, the ICE needs to be turned on and the controller moves to the *ICE Transition* mode. The driver torque request in this mode is met by the EM as the ICE is not yet turned on. If the condition that the driver torque request is greater than 50 Nm persists for 0.5 seconds, or if the driver torque request exceeds 50% of the EM maximum axle torque, the controller enters the *Pre-ICE Transition* state. If the drive torque request falls below 0, the controller moves back to the *Coasting* mode. In the *Pre-ICE Transition* state, outputs are exactly the same as *ICE Transition* state except that

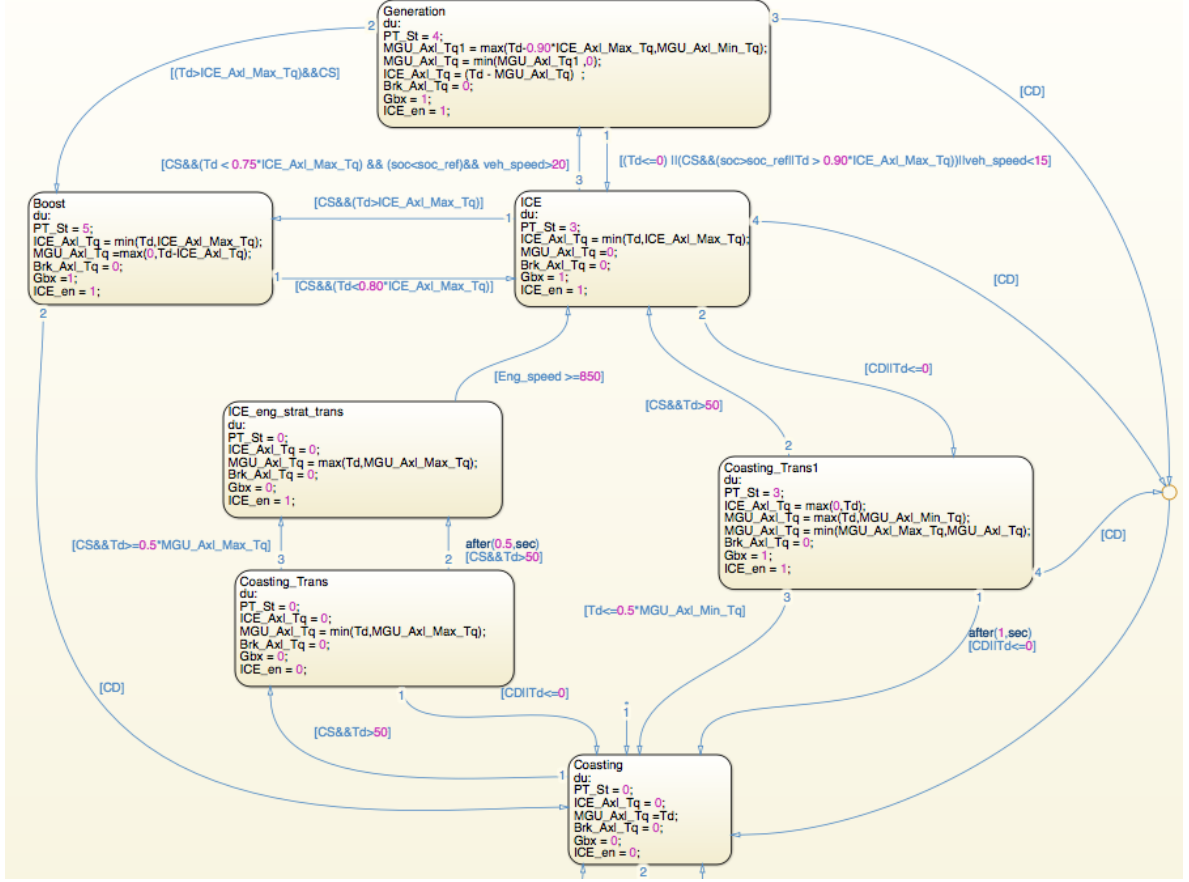


Figure 35: Coasting and CS States

the engine enable flag is set to 1. When the engine speed exceeds 850 RPM, i.e., when the engine is idling, the controller enters the *ICE* mode. In this mode, the gearbox is enabled. The driver torque request is met by the ICE and the EM torque is set to 0.

When the controller is in the *ICE* mode and the driver torque request exceeds the ICE maximum axle torque, the controller moves to the *Boost* mode. When *Boost* mode is active, the controller moves back to the *ICE* mode only when the driver torque request falls below 80% of the ICE maximum axle torque. Also, when *Boost* mode is active, the EM assists the ICE only when the driver torque request exceeds the ICE maximum axle torque

When the controller is operating in *ICE* mode, it moves to the *Generation* mode



when the vehicle speed is greater than 20 MPH, the SOC is less than the reference SOC and the driver torque request falls below 75% of the ICE maximum axle torque. The goal here is to move the ICE close to a full-load condition so that it operates in a more efficient region. This is done by operating the EM in the generating region. The controller moves from the *Generation* mode to *ICE* mode if one of the following conditions is satisfied:

- Vehicle speed is less than 15 MPH
- The driver torque request exceeds 90% of the ICE maximum axle torque
- The SOC exceeds the reference SOC
- The driver torque request falls below 0

When the controller is in *Generation* mode, the EM torque is set such that the ICE load is 90%. If the driver torque request falls below 0, the controller moves to the *Coasting Transition* state. The outputs of the *Coasting Transition* state are set such that an accelerating driver torque request is met by the ICE and any decelerating torque request is met by the EM. If the condition that the driver torque request falls below 0 persists for 1 sec or if the driver torque request falls below 50% of the EM axle minimum torque, then the control moves to *Coasting* mode. Else if the driver torque demand exceeds 0, the controller moves back to the *ICE* mode.

The states that control the braking of the vehicle are shown in Figure 36.

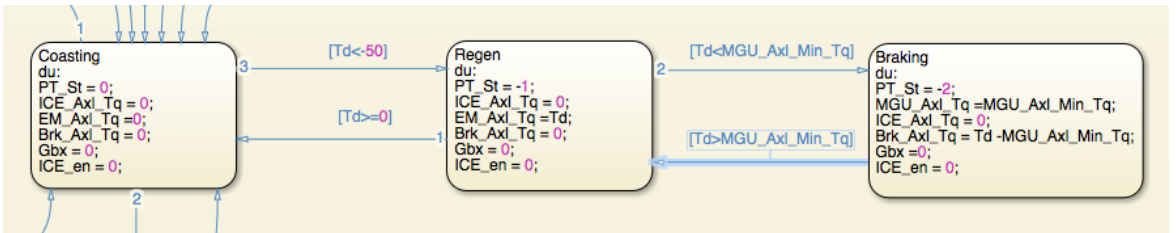


Figure 36: Coasting and Braking states

Table 5: Basic Powertrain Controller Powertrain States

$PT\_st$	State
-2	Braking
-1	Regeneration
0	Coasting
1	Electric
2	Dual
3	ICE
4	Generation
5	Boost

When the controller is in *Coasting* mode and the driver torque request falls below -50 Nm, the controller moves to *Regeneration* mode where the decelerating driver torque request is met completely by the EM. If the driver torque request becomes positive, the controller moves back to *Coasting* mode. When in *Regeneration* mode, if the driver torque request falls below the EM minimum axle torque, the controller moves to *Braking* mode. If the driver torque request exceeds the EM minimum axle torque, the controller moves back to the *Regeneration* mode. In the *Braking* mode, the EM provides maximum generating torque and the brakes are used to meet the driver torque demand.

The operation of the TCM and the gear shift strategy are as described in Appendix A. The different values of  $PT\_st$  correspond to the powertrain states as given in Table 5. The transition states take the state numbers of the main states they emerge from.

#### 4.6 Charge Depletion Mode Simulation

The graphs for the simulation where the vehicle is driven in CD mode for the US06 driving cycle are shown below. The vehicle speed is shown in Figure 37. The ICE and EM speeds are shown in Figure 38. ICE and EM torques are shown in Figure

39. The powertrain state, battery SOC and gear number are shown in Figure 40.

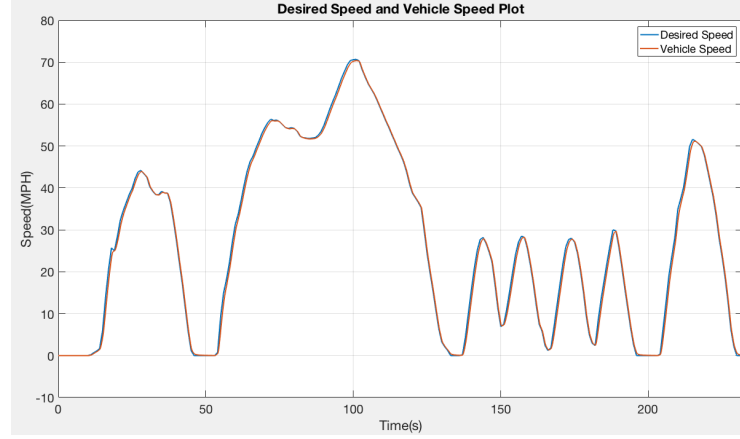


Figure 37: CD Mode Desired Speed and Vehicle Speed

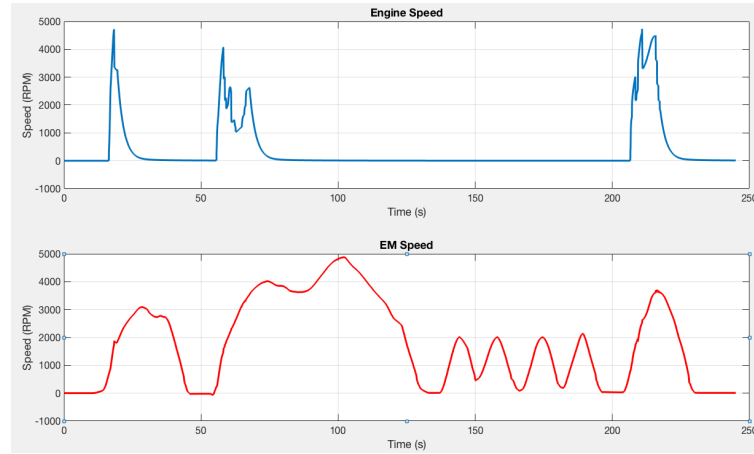


Figure 38: CD Mode Engine and Motor Speed

#### 4.7 Charge Sustain Mode Simulation

The graphs for the simulation where the vehicle is driven in CS mode for the US06 driving cycle are shown below. The vehicle speed is shown in Figure 41. The ICE and EM speeds are shown in Figure 42. ICE and EM torques are shown in Figure 43. The powertrain state, battery SOC and gear number are shown in Figure 44.

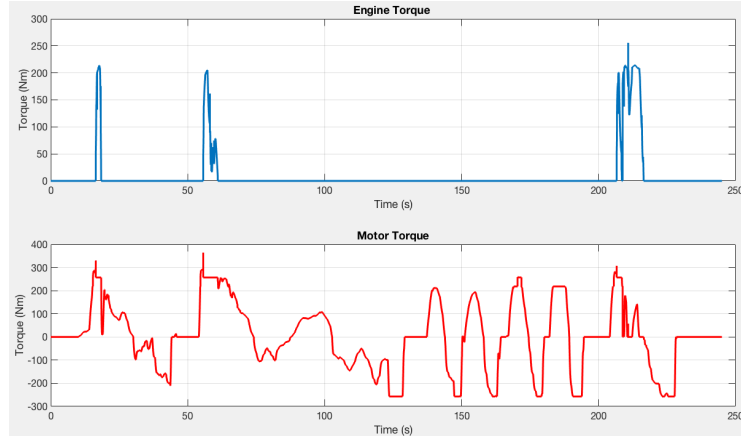


Figure 39: CD Mode Engine and Motor Torque

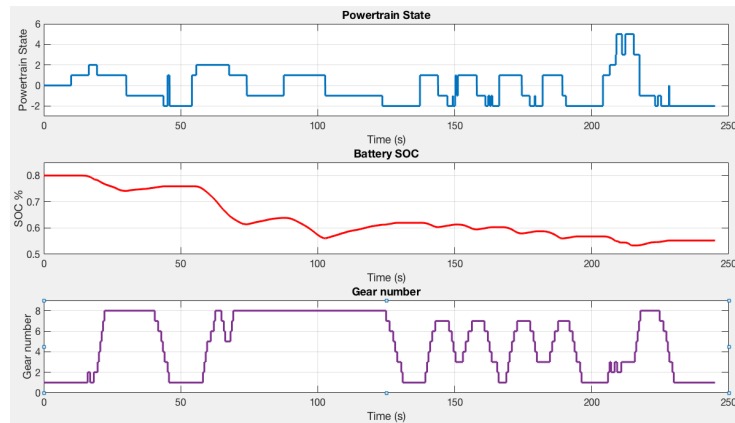


Figure 40: CD Mode Powertrain State, Battery SOC and Gear Number

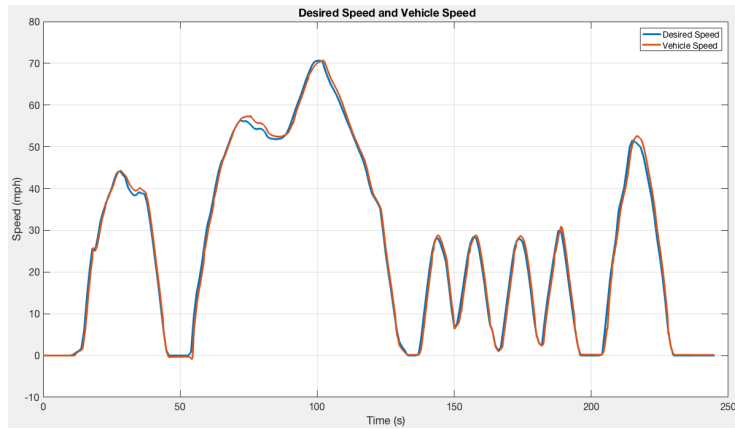


Figure 41: CS Mode Desired Speed and Vehicle Speed

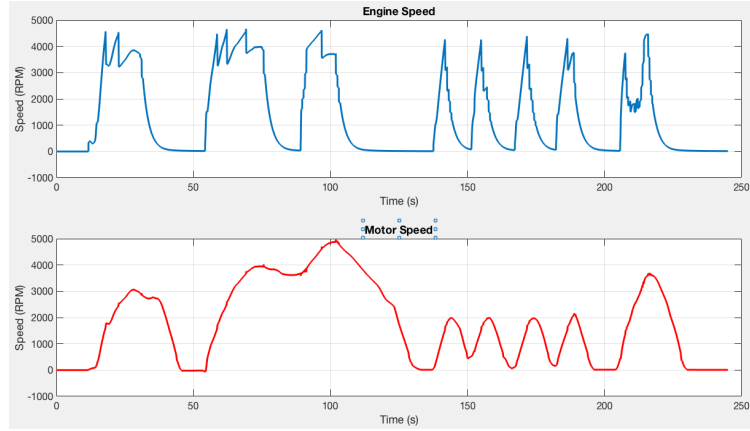


Figure 42: CS Mode Engine and Motor Speed

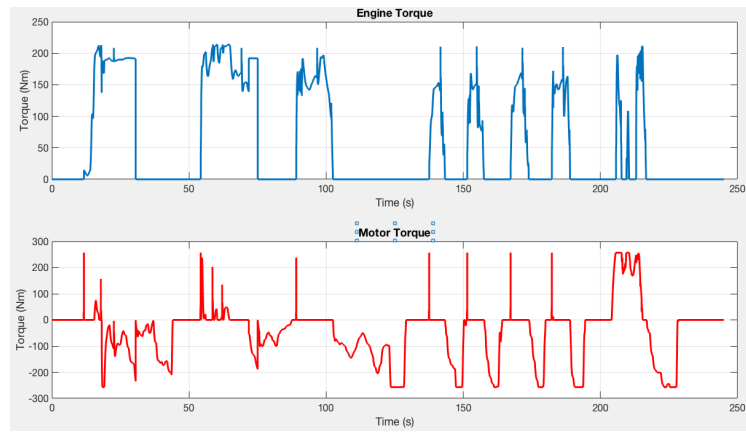


Figure 43: CS Mode Engine and Motor Torque

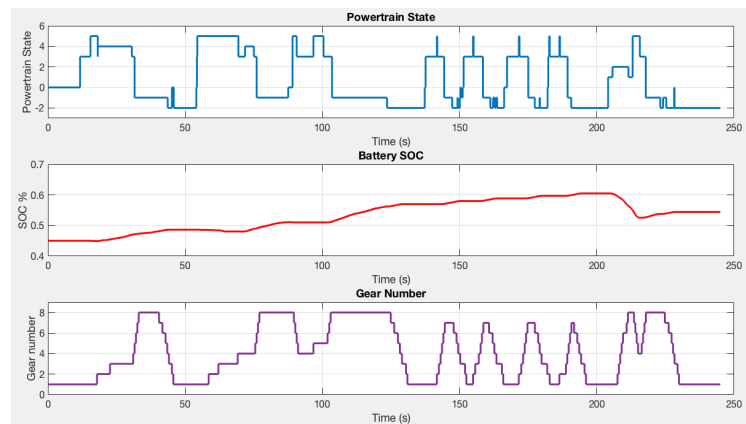


Figure 44: CS Mode Powertrain State, Battery SOC and Gear Number

It can be seen that the vehicle moves from CS mode to CD mode at 213 seconds, when the SOC reaches 0.6.

## CHAPTER V

### EQUIVALENT CONSUMPTION MINIMIZATION STRATEGY

Equivalent consumption minimization strategy is an optimal control strategy for a hybrid electric vehicle developed using Pontryagin's minimum principle. The goal is to minimize the total fuel energy consumption, which is given by the product of the lower heating value of the fuel  $H_l$  and the fuel mass flow  $\dot{m}_f$  integrated over the total time of vehicle operation. Hence, the total fuel energy consumption is taken as the cost function  $J(u(t))$  and is expressed as:

$$J = \int_0^{t_f} H_l \dot{m}_f dt \quad (36)$$

The only state variable considered is the battery SOC and the state dynamics are given by:

$$\dot{\xi} = -\frac{I_{BT}}{Q_0} \approx -P_b - \beta P_b^2 \quad (37)$$

where  $P_b$  is the battery power delivered to the electric motor and  $\beta = R_i/V_{OC}^2$ . Hence a Hamiltonian can be developed using the cost function and considering the state dynamics as the constraint. The costate associated with the battery dynamics is scalar and is substituted by a dimensionless value called the equivalence factor to convert the state dynamics into total electric power consumption. This substitution results in the Hamiltonian being equal to the sum of total fuel power and a weighted electric power as shown in [16]. Hence the Hamiltonian can be viewed as the equivalent vehicle power consumption, thus lending the name *equivalent consumption minimization strategy* to this method.

The optimal EM power that minimizes the Hamiltonian for a given driver power demand is taken as the plant input. The ICE power is calculated as the difference between the driver power demand and the optimal EM power. Implementation of the control strategy in this form requires the real-time minimization of the Hamiltonian at every time step. This is computationally expensive and renders the control strategy inapplicable for application in real vehicles.

The author of [16] presents an explicit control law that is derived from the above strategy. The control law is just a function of the vehicle parameters and the equivalence factor. This does not require any real-time minimization or optimization. This strategy is thus computationally inexpensive and implementable in real vehicles. In this chapter, this control law is applied to the P3 parallel hybrid vehicle developed in Chapter 2 and the vehicle performance is evaluated. The gear selection strategy as described in Appendix A is retained for the ECMS simulations.

## 5.1 *ECMS Overview*

### 5.1.1 Equivalence Factor

The block diagram of the equivalent consumption minimization strategy (ECMS) structure is shown in Figure 45. The control law presented in [16] also penalizes and considers the deviation of the battery SOC  $\xi$  from a reference SOC  $\xi_{ref}$  in the Hamiltonian. The fuel energy required to return the final battery SOC at the end of the drive cycle, to the reference SOC is viewed as the cost-to-go. The equivalence factor  $s$  is calculated as:

$$s(\xi(t)) = s_0 + \int_0^t \frac{\xi_{ref} - \xi(\tau)}{T_i} d\tau + \frac{\tilde{\alpha}}{Q_0 V_{OC}(\xi(t))} \cdot (\xi_{ref} - \xi(t))^{2q-1} \quad (38)$$

where

$$\tilde{\alpha} = \frac{2q\alpha T_h}{(1 + 2q)\Delta\xi_{nrm}^{2q}} \quad (39)$$



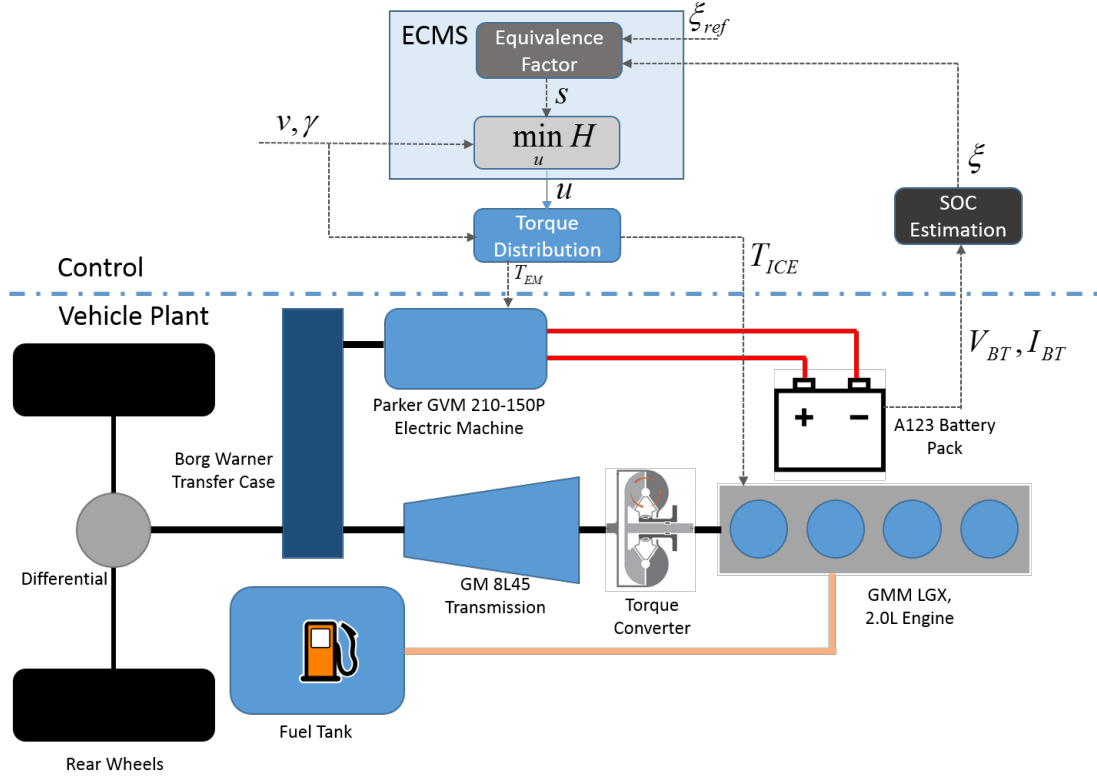


Figure 45: ECMS Structure

Here,  $\alpha$  is a weighting parameter,  $\Delta\xi_{norm}$  normalizes the SOC deviation and  $q \in N$  gives the order of the penalty. In the cost-to-go,  $T_h$  is the time period considered to return the SOC at the end of the drive cycle to the reference SOC. The equivalence factor is initialized with the value  $s_0$ . An integrator with integration time  $T_i$  is used to account for the changes in conversion efficiency over time.

### 5.1.2 Control Strategy

The author of [16] uses a simplified powertrain model as shown in Figure 46, to derive an explicit control law from Pontryagin's Minimum Principle. For this simplified model,  $P_{ICE}$  is the ICE power,  $P_{EM}$  is the EM power,  $P_f$  is the fuel power and  $P_d$  is the drive power demand. Initially, the inputs are assumed to be unconstrained and the component efficiencies are assumed to be constant. The Hamiltonian for this

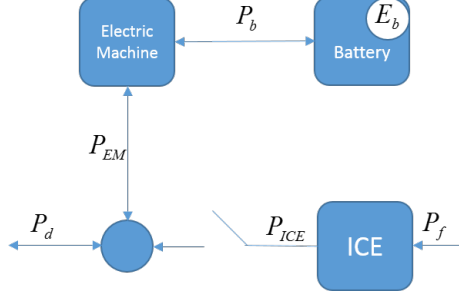


Figure 46: Simplified Vehicle Plant Model

model is discontinuous and each piece-wise continuous region represents a different hybrid operating mode. The Hamiltonian for this model is given by:

$$H = P_f - \lambda \cdot (P_b + \beta P_b^2) \quad (40a)$$

$$= \frac{P_d - P_{EM} + P_0}{e} \cdot B + s(P_{EM} \cdot \eta^{-\text{sign}(P_{EM})} + \beta P_{EM}^2 \cdot \eta^{-2\text{sign}(P_{EM})}) \quad (40b)$$

where  $e$  is the ICE internal efficiency,  $P_0$  is the ICE friction power,  $\eta$  is the EM efficiency and  $B$  is a binary value that represents engine on/off state. The costate  $\lambda$  in Equation 40a is replaced by  $-s$  in Equation 40b. The Hamiltonian is explicitly minimized in each region by differentiating the piece-continuous function by the EM power. This gives explicit expressions for the optimal EM power in each operating region (Appendix D). Evaluation of the Hamiltonian at the points of discontinuity gives the expressions for the boundaries of each operating mode (given in Appendix D). The expressions for the optimal EM power and the boundaries for the operating modes do not change when considering input constraints and dynamic component efficiencies. A reader interested in the complete derivation of the control law is directed to Chapter 5 of [16].

The control strategy thus obtained has seven operating modes: *Electric*, *ICE*, *Boost*, *Generation*, *ICE Max*, *Regenerative Braking* and *Mechanical Braking*. Similar to the Basic Supervisory Controller, in the *Electric* mode, the vehicle is powered only by the EM and the ICE is turned off. In the *ICE* mode, the vehicle is powered only by the ICE. In *Boost* mode, the EM and the ICE both provide torque to the wheels. In

*Generation* mode, the EM is operated in the generating region to recharge the battery and the ICE supplies the driver power demand plus the EM power. In the *ICE Max* mode, the ICE operates at maximum power and the EM is either in motoring mode or generating mode depending on the value of equivalence factor.

The control law identifies two values of the equivalence factor:  $s_{Boost}$  and  $s_{Gen}$ . The equations used to determine these values are given in the Appendix D. The vehicle is allowed to operate in *Boost* mode only when the equivalence factor is less than  $s_{Boost}$  and the vehicle is allowed to operate in *Generation* mode only if the equivalence factor is greater than  $s_{Gen}$ . The control law also prescribes certain power threshold values which are used along with  $s_{Boost}$  and  $s_{Gen}$  to determine the boundaries of operating modes.

- When  $s > s_{Gen}$ , the vehicle operates in *Electric* mode if the driver power demand  $P_d$  is less than the threshold  $P_{lim}^e$ .
- When  $s > s_{Gen}$ , the vehicle transitions from *Electric* mode to *Generation* mode if  $P_d$  is greater than the threshold  $P_{lim}^e$ .
- When  $s > s_{Gen}$ , the vehicle transitions from *Generation* mode to *ICE Max* mode if  $P_d$  is greater than  $P_{ICE_{Max}} + P_{EM}^e$ .
- When  $s < s_{Boost}$ , the vehicle operates in *Electric* mode if  $P_d$  is less than the threshold calculated as the minimum of  $P_{lim}^{bo}$  and  $P_{EM_{Max}}$ .
- When  $s < s_{Boost}$ , the vehicle transitions from *Electric* mode to *Boost* mode if  $P_d$  is greater than the threshold calculated as the minimum of  $P_{lim}^{bo}$  and  $P_{EM_{Max}}$ .
- When  $s < s_{Boost}$ , the vehicle transitions from *Boost* mode to *ICE Max* mode if  $P_d$  is greater than  $P_{ICE_{Max}} + P_{EM}^{bo}$ .

- When  $s_{Gen} < s < s_{Boost}$ , the vehicle operates in *Electric* mode if  $P_d$  is less than the threshold  $P_{lim}^{th}$ .
- When  $s_{Gen} < s < s_{Boost}$ , the vehicle transitions from *Electric* mode to *ICE* mode if  $P_d$  is greater than the threshold  $P_{lim}^{th}$ .
- When  $s_{Gen} < s < s_{Boost}$ , the vehicle transitions from *ICE* mode to *ICE Max* mode if  $P_d$  is greater than  $P_{ICE_{Max}}$ .

The equations used to calculate  $P_{lim}^{re}$ ,  $P_{lim}^{bo}$ ,  $P_{lim}^{th}$ ,  $P_{EM}^{re}$  and  $P_{EM}^{bo}$  are given in Appendix D. The *Regenerative Braking* and *Mechanical Braking* modes are implemented as done in the Basic Supervisory Controller. For this causal non-predictive ECMS, the reference SOC is kept at a constant value of 0.55. This value is chosen as it is the mean between minimum and maximum SOC values, which are 0.3 and 0.8 respectively.

As the control strategy is also a function of the vehicle speed, the power thresholds are converted to torque thresholds to easily visualize the control strategy. As the total axle torque available is also dependent on the gear as shown in Figure 33, the regions that define the optimal control also differ for each gear. As examples, the optimal control regions for gear 1 when the equivalence factor is 3.5 and 4.05 are shown in Figures 47 and 48 respectively. The Y-axis is the driver torque demand calculated as the quotient of the driver power demand  $P_d$  and the axle speed  $w_{axl}$ .

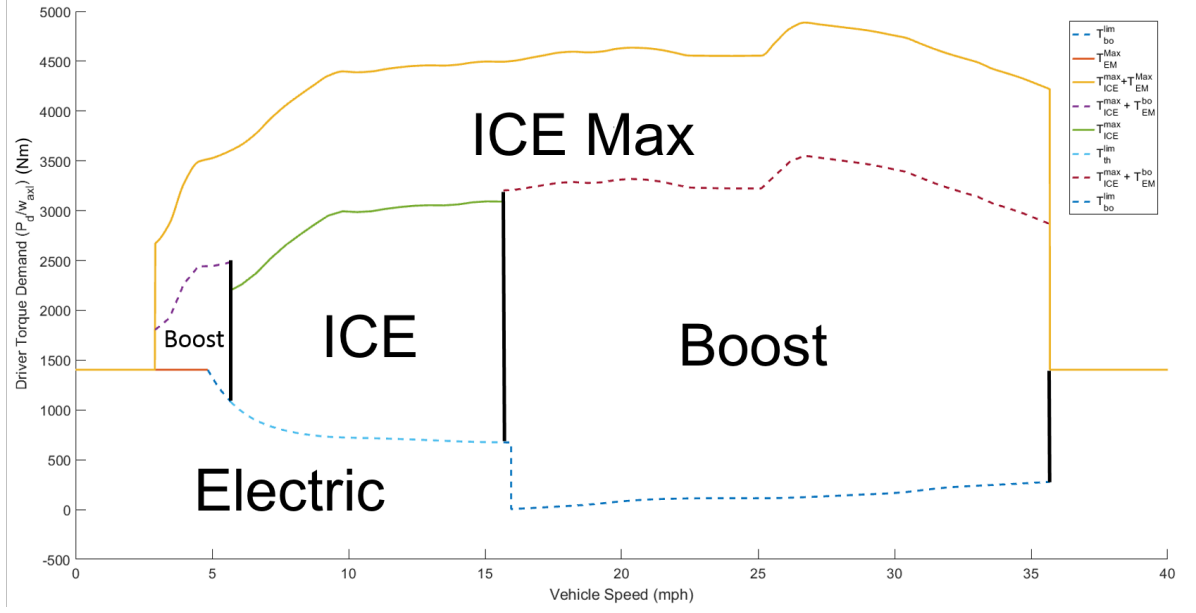


Figure 47: Optimal Control Regions at Gear 1 when Equivalence Factor is 3.5

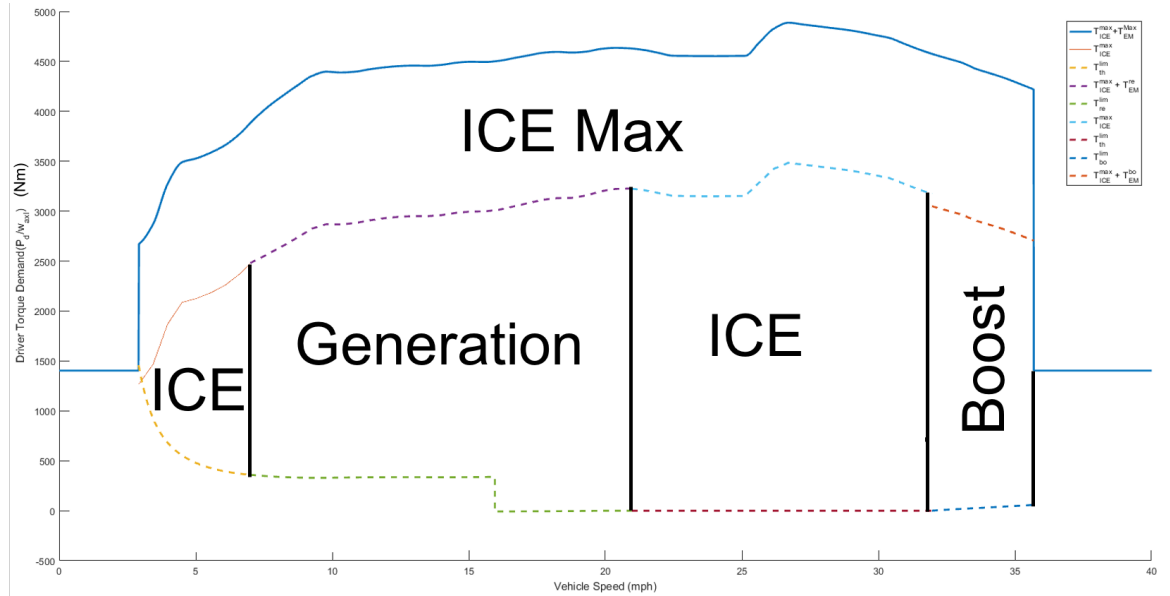


Figure 48: Optimal Control Regions at Gear 1 when Equivalence Factor is 4.05

A block diagram of the controller is given in Figure 49. The Causal Feedback Controller (CFC) block is used to calculate the equivalence factor  $s$  using Equation 38. Using the equivalence factor and the driver power demand, a Selective Hamiltonian

Minimization (SHM) is done to choose the appropriate control candidates enumerated in Equation 42 given in Appendix D. The parameters used for the feedback law are given in Table 6.

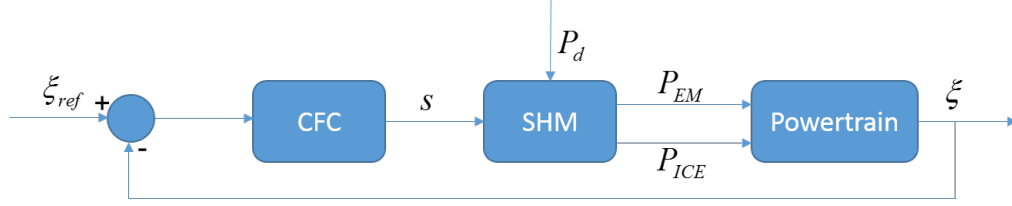


Figure 49: Causal Feedback Controller

Table 6: Parameters for Causal Feedback Control

Parameter	Value	Unit
$s_0$	3.1	-
$T_i$	100	s
$\xi_{ref}$	0.55	-
$\Delta\xi_{nrm}$	0.1	-
$q$	2	-
$T_h$	500	s
$\alpha$	2000	W
$Q_0$	8	Ah
$V_{OC}$	324	V

The value for  $s_0$  that ensures charge sustenance within a tolerance of 0.002 ( $\xi_0 = \xi_f = 0.55$ ) was iteratively obtained by testing the controller for different values of  $s_0$ . The charge sustaining values for the US06 City, US06 Highway, HWFET and US505 drive cycles were found to be 3.075, 2.97, 3.017 and 3.087 respectively. Prioritizing city driving and rounding off, a value of 3.1 for  $s_0$  was chosen for all drive cycles.

## 5.2 Simulink Structure for ECMS Supervisory Control

The component limits are calculated just as they were in the Basic Supervisory Controller. Here, apart from the torque limits, the power limits are also calculated. This is shown in Figures 50 and 51. The implementations for the determination of component efficiencies and ICE friction power are shown in Figure 52.

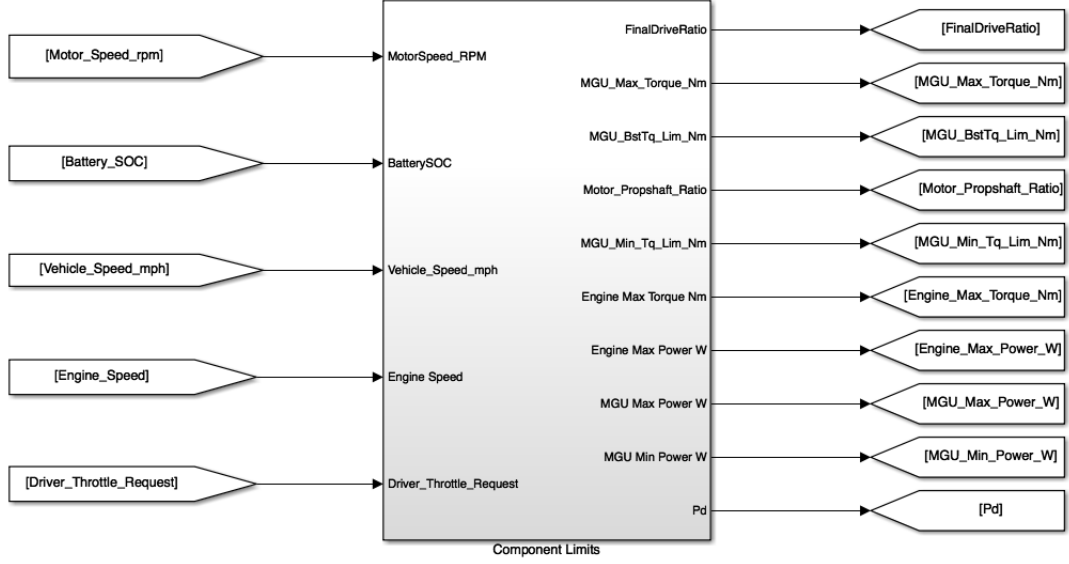


Figure 50: Component Limits Block

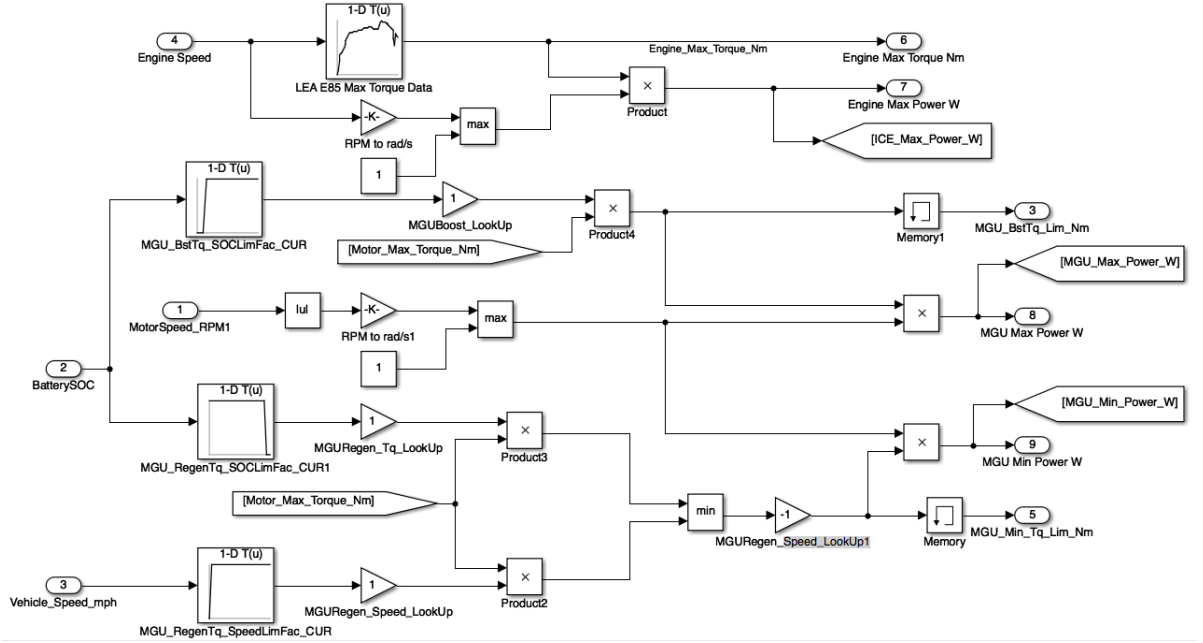


Figure 51: Component Limits Calculations

The average ICE, EM and the engine friction power are calculated as a function of ICE and EM speeds using look-up tables. The equivalence factor is calculated as given in Equation 38. This is shown in Figure 53. The power threshold values for mode transitions  $P_{EM}^{re}$ ,  $P_{EM}^{bo}$ ,  $P_{lim}^{bo}$ ,  $P_{lim}^{th}$  and  $P_{lim}^{re}$  are done according to equations

given in Appendix D. Only an overview of the block where the calculations are done is shown in Figure 54. The detailed implementations of these calculations are shown in Appendix E. Values of  $s_{Boost}$  and  $s_{Gen}$  are calculated in Figure 54. These values are then used to select the power limits applicable to the current operating mode.

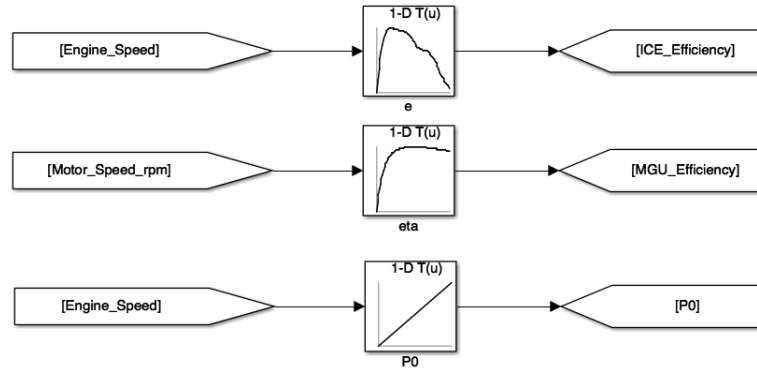


Figure 52: Component Efficiencies

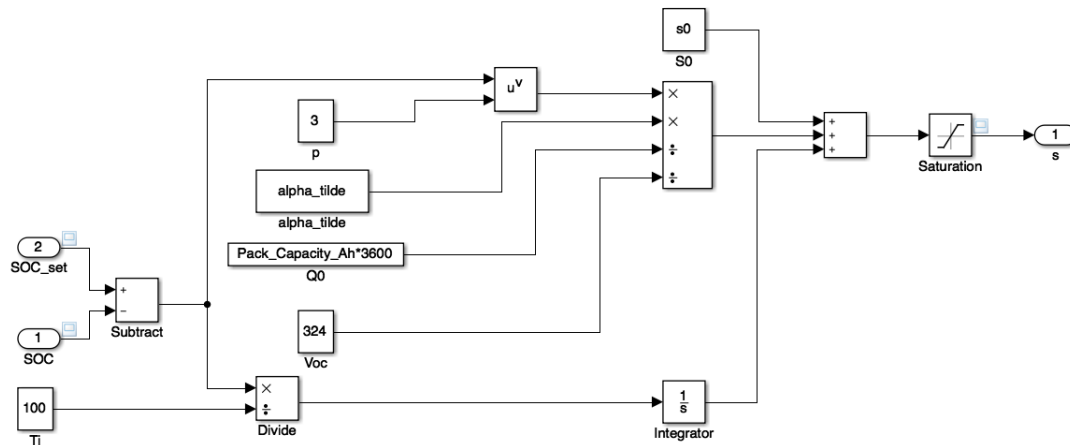


Figure 53: Calculation of Equivalence Factor



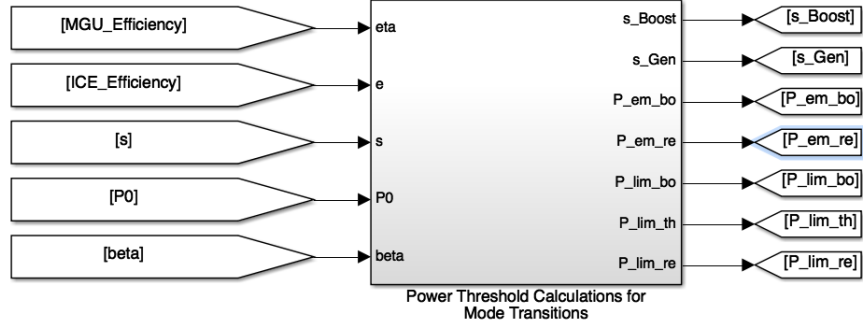


Figure 54: Overview of Power Threshold Calculations for Mode Transitions

If the driver power demand is lower than  $P_{lowthresh}$ , the controller transitions into *Electric* mode. Hence

- If  $s < s_{Boost}$ ,  $P_{lowthresh}$  is  $\min(P_{lim}^{bo}, P_{EM_{max}})$ .
- If  $s_{Boost} < s < s_{Gen}$ ,  $P_{lowthresh}$  is  $P_{lim}^{th}$ .
- If  $s_{Gen} < s$ ,  $P_{lowthresh}$  is  $P_{lim}^{re}$ .

If the driver power demand is higher than  $P_{upthresh}$ , the controller transitions into *ICE Max* mode. Hence

- If  $s < s_{Boost}$ ,  $P_{upthresh}$  is  $\min(P_{EM}^{bo} + P_{ICE_{max}}, P_{EM_{max}} + P_{ICE_{max}})$ .
- If  $s_{Boost} < s < s_{Gen}$ ,  $P_{upthresh}$  is  $P_{ICE_{max}}$ .
- If  $s_{Gen} < s$ ,  $P_{upthresh}$  is  $P_{ICE_{max}} + P_{EM}^{re}$ .

The inputs and outputs to the subsystem where these calculations are done are shown in Figure 55. The implementation of the calculations is shown in Figure 56.

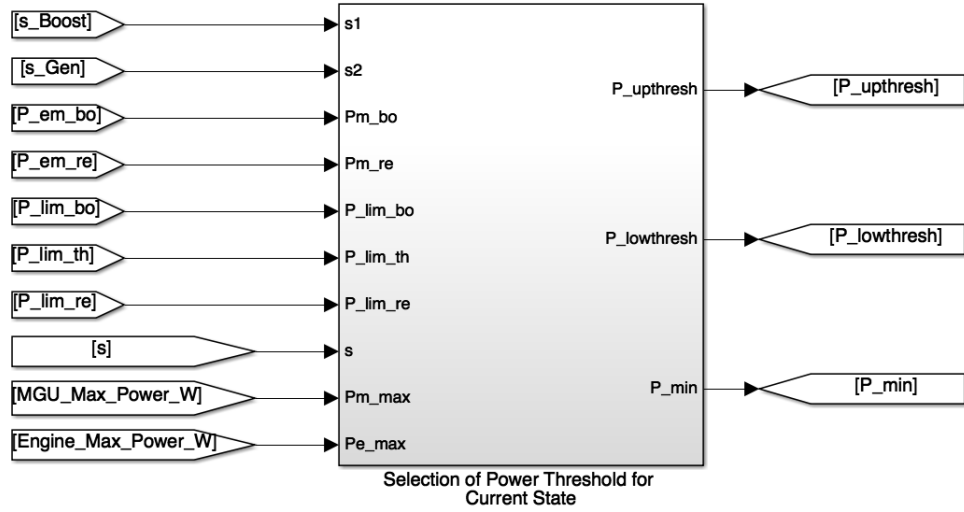


Figure 55: Overview of Power Threshold Selection for Current Mode

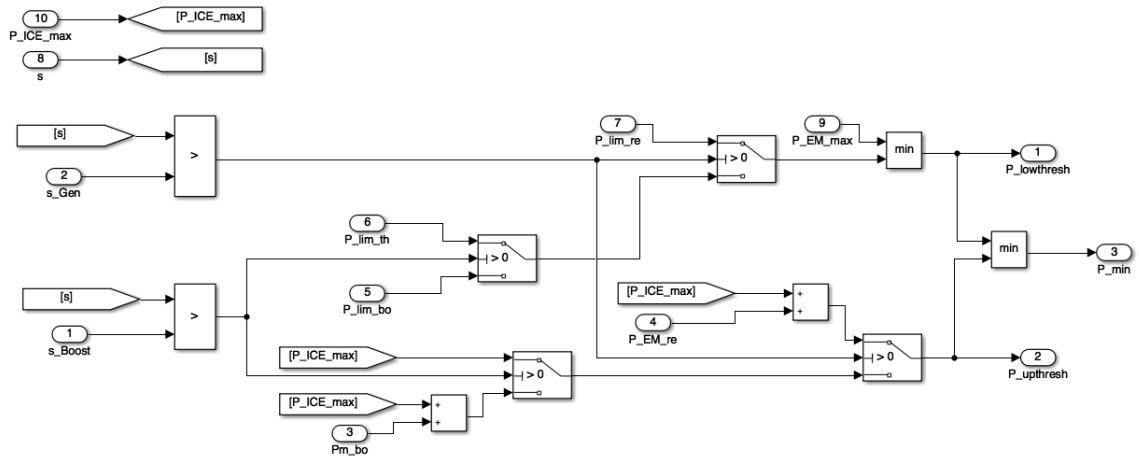


Figure 56: Calculations of Power Threshold Selection for Current Mode

A truth table is used to evaluate the conditions to transition into any state. This is shown in Figures 57, 58 and 59.

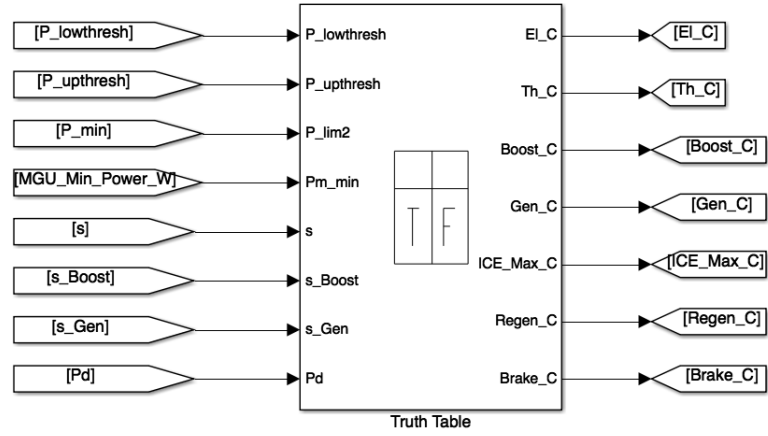


Figure 57: Truth Table Ports

	Description	Condition	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
1	Power Condition	$P_{lowthresh} < P_{upthresh}$	T	T	T	T	T	-	-	F	F	T	T	-
2	Power Condition	$P_d > P_{lim2}$	-	-	-	-	-	-	F	T	-	-	-	-
3	Power Condition1	$P_d > P_{lowthresh}$	F	T	T	T	T	-	-	-	-	F	T	-
4	Power Condition2	$P_d > P_{upthresh}$	F	F	F	F	T	-	-	-	-	T	F	-
5	Power Condition2	$P_d < 0$	F	F	F	F	F	T	T	F	F	F	F	F
6	Power Condition2	$P_d < P_{m\_min}$	F	F	F	F	F	F	T	F	F	F	F	T
7	s condition	$s > s_{Boost}$	-	T	F	T	-	-	-	-	-	-	F	-
8	s condition	$s > s_{Gen}$	-	F	F	T	-	-	-	-	-	-	T	-
		Actions: Specify a row from the Action Table	1	2	3	4	5	6	7	1	5	5	4	7

Figure 58: Truth Table Conditions

#	Description	Action
1	Transition to Electric Drive Valid	El_C =1>0; Th_C = 1<0; Boost_C = 1<0; Gen_C =1<0; ICE_Max_C =1<0; Regen_C =1<0; Brake_C = 1<0;
2	Transition to ICE Mode Valid	El_C =1<0; Th_C = 1>0; Boost_C = 1<0; Gen_C =1<0; ICE_Max_C =1<0; Regen_C =1<0; Brake_C = 1<0;
3	Transition to Boost Mode Valid	El_C =1<0; Th_C = 1<0; Boost_C = 1>0; Gen_C =1<0; ICE_Max_C =1<0; Regen_C =1<0; Brake_C = 1<0;
4	Transition to Generation Mode Valid	El_C =1<0; Th_C = 1<0; Boost_C = 1<0; Gen_C =1>0; ICE_Max_C =1<0; Regen_C =1<0; Brake_C = 1<0;
5	Transition to ICE_Max Valid	El_C =1<0; Th_C = 1<0; Boost_C = 1<0; Gen_C =1<0; ICE_Max_C =1>0; Regen_C =1<0; Brake_C = 1<0;
6	Transition to Regen Braking Valid	El_C =1<0; Th_C = 1<0; Boost_C = 1<0; Gen_C =1<0; ICE_Max_C =1<0; Regen_C =1>0; Brake_C = 1<0;
7	Transition to Mech Braking Valid	El_C =1<0; Th_C = 1<0; Boost_C = 1<0; Gen_C =1<0; ICE_Max_C =1<0; Regen_C =1<0; Brake_C = 1>0;

Figure 59: Truth Table Actions

The realization of the various operating modes of the ECMS strategy is done using a Stateflow chart. The chart takes the transition conditions generated by the truth table, the speed dependent component efficiencies and the component power limits as inputs. The outputs of the state machine are the powertrain state, the ICE power request, EM request, mechanical brake power request and the enable flags for the gearbox and ICE. The state model is split into three Figures 60, 61 and 62 due to the size of the structure.

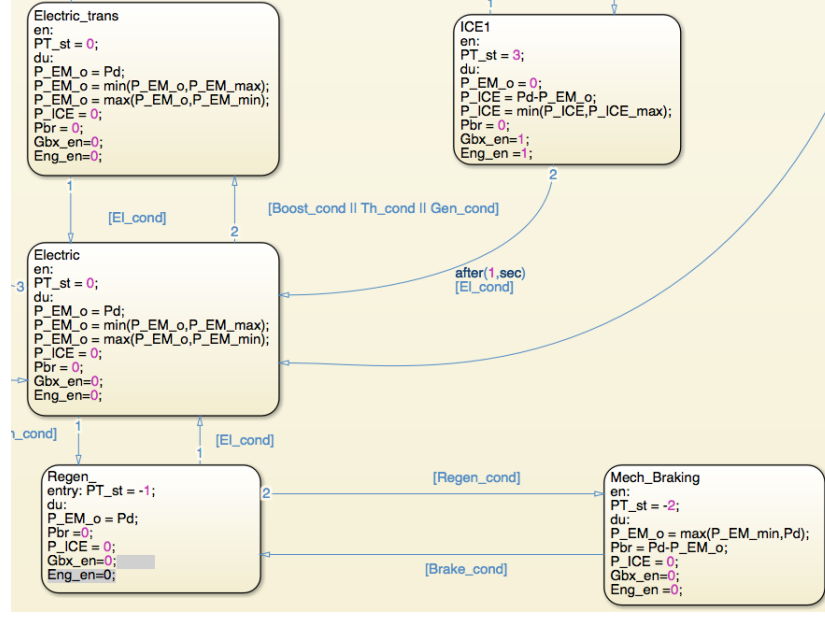


Figure 60: ECMS State Machine Part 1

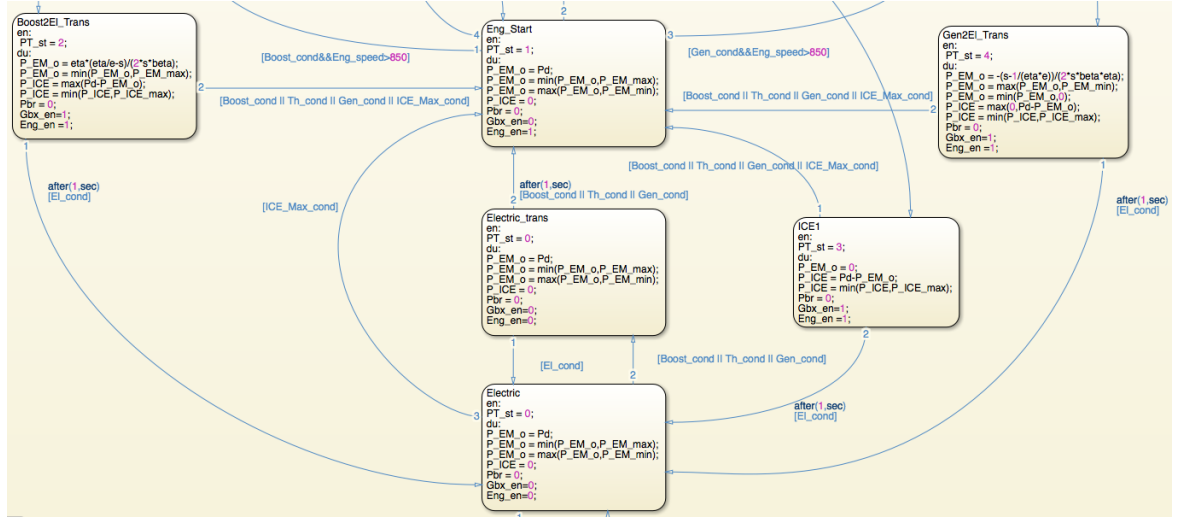


Figure 61: ECMS State Machine Part 2

The optimal motor power in each state is calculated according to Equation 43 given in Appendix D. The ICE needs to be started and the gearbox needs to be engaged when transitioning from *Electric* mode to *Boost*, *Thermal (ICE)*, *Generation* or *ICE Max* modes. Hence, whenever the control is in *Electric* mode and the conditions to transition to any of these states are true, the control first moves to the *Electric Transition* state. If any one of the conditions persists for more than a second, the

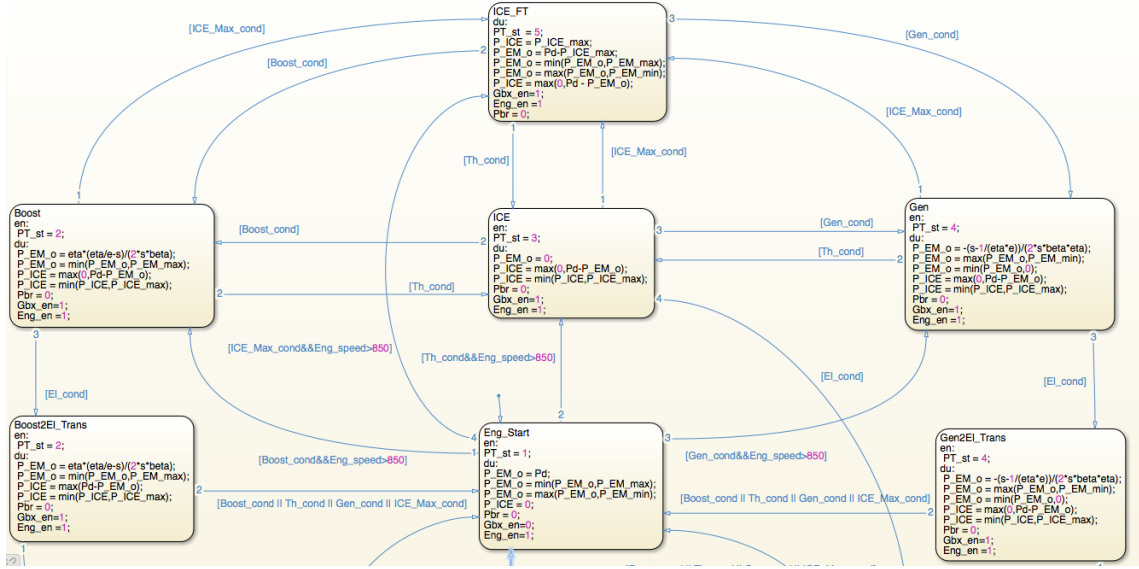


Figure 62: ECMS State Machine Part 3

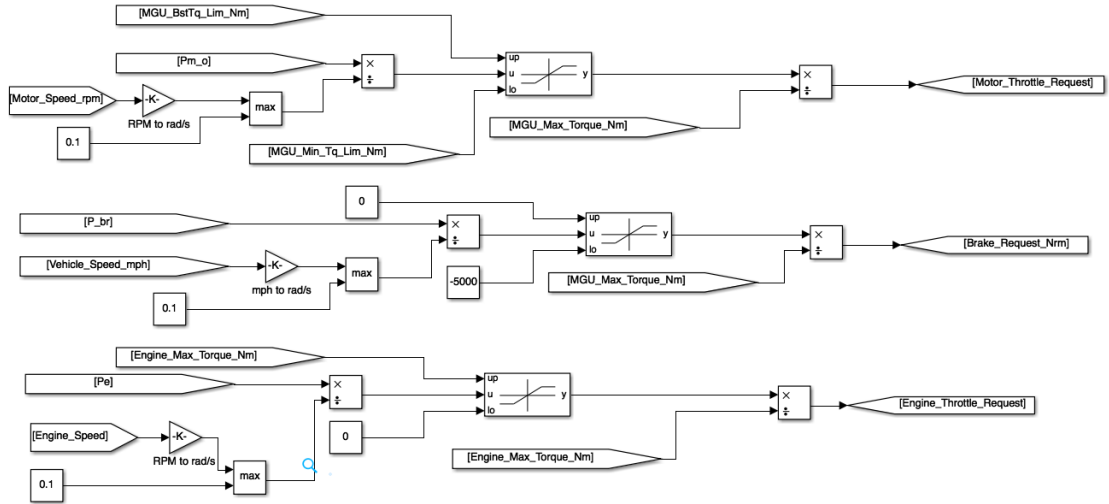


Figure 63: ECMS Power to Torque Conversion

control moves to *Engine Start* where the ICE enable flag is set. Once the ICE reaches idling speed, the control moves to *Boost*, *Thermal*, *Generation* or *ICE Max* mode depending on which transition condition is true. A direct transition from *ICE Max* mode to *Electric* mode is not allowed. The control needs to go through either *Boost*, *Thermal* or *Generation* mode.

When the control needs to move from *Boost*, *Thermal* or *Generation* modes back

to the *Electric* mode, the ICE needs to be turned off and the gearbox needs to be disengaged. Hence a transition state is introduced between each of these states and the *Electric* mode to prevent the ICE from turning on and off too often. The control moves to the *Electric* mode only if the condition persists for 1 second.

Regenerative braking and mechanical braking are handled just like the Basic Supervisory Controller. The conversion of the individual component power requests to normalized torque requests is shown in Figure 63.

### **5.3 *Simulation Results***

#### **5.3.1 Comparison of ECMS to Basic Supervisory Controller**

Simulations for the ECMS were done for the US06 City, US06 Highway, US505 and HWFET drive cycles without considering changes in road gradient. The simulations are done setting the initial and reference SOC to 0.55. The minimum and maximum allowed SOC limits are set to 0.3 and 0.8 respectively. The equivalence factor is computed with parameters given in Table 6. The values that  $PT\_st$  takes correspond to states in the ECMS as given in Table 7.

To compare the performance of the ECMS with the Basic Supervisory Controller, simulations of the Basic Supervisory Controller were also carried out with the same conditions as that for the ECMS, for all the drive cycles.

A plot of the battery SOC for the US505 drive cycle is shown in Figure 64. A plot of the corresponding powertrain state is shown in Figure 65. The SOC plots for the Basic Supervisory Controller and the ECMS are shown in Figures 66 to 69.

Table 7: ECMS Powertrain States

$PT\_st$	State
-2	Braking
-1	Regenerative Braking
0	Electric
1	Engine Start
2	Boost
3	ICE
4	Generation
5	ICE Max

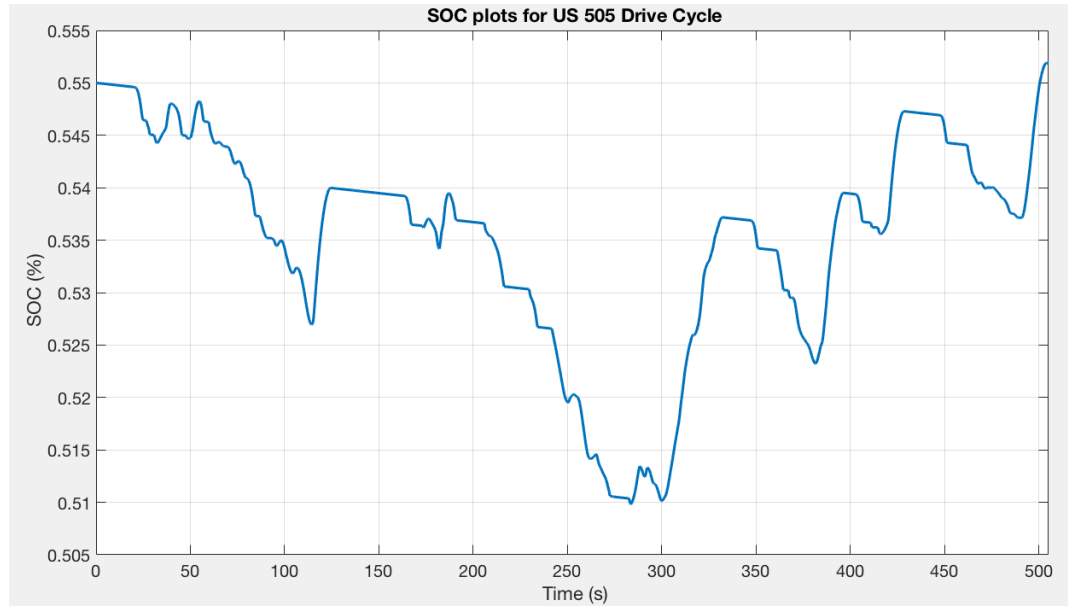


Figure 64: SOC Plot for ECMS Simulation for US505 Drive Cycle



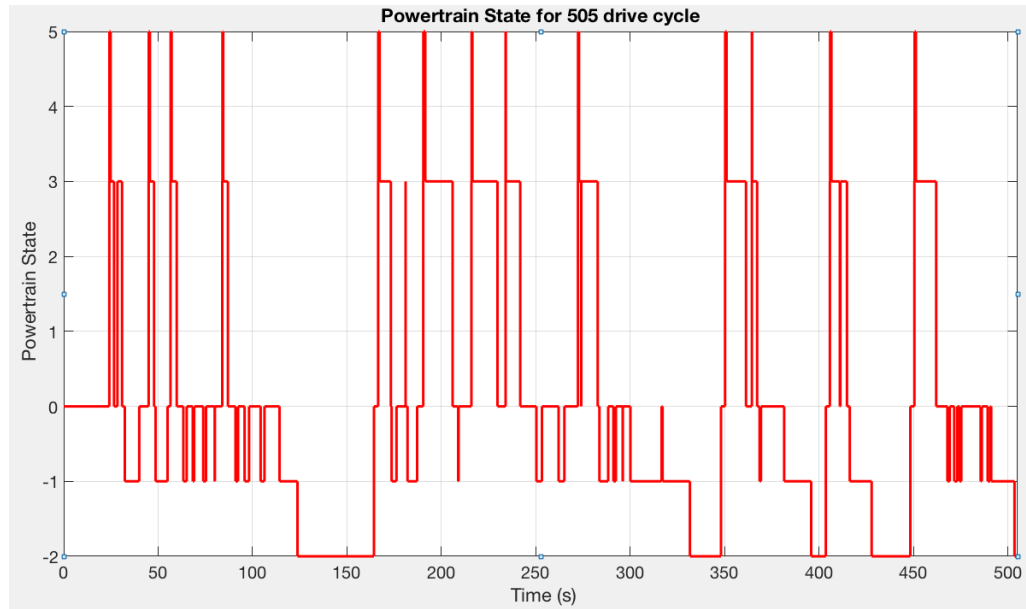


Figure 65: Powertrain State Plot for ECMS Simulation for US505 Drive Cycle

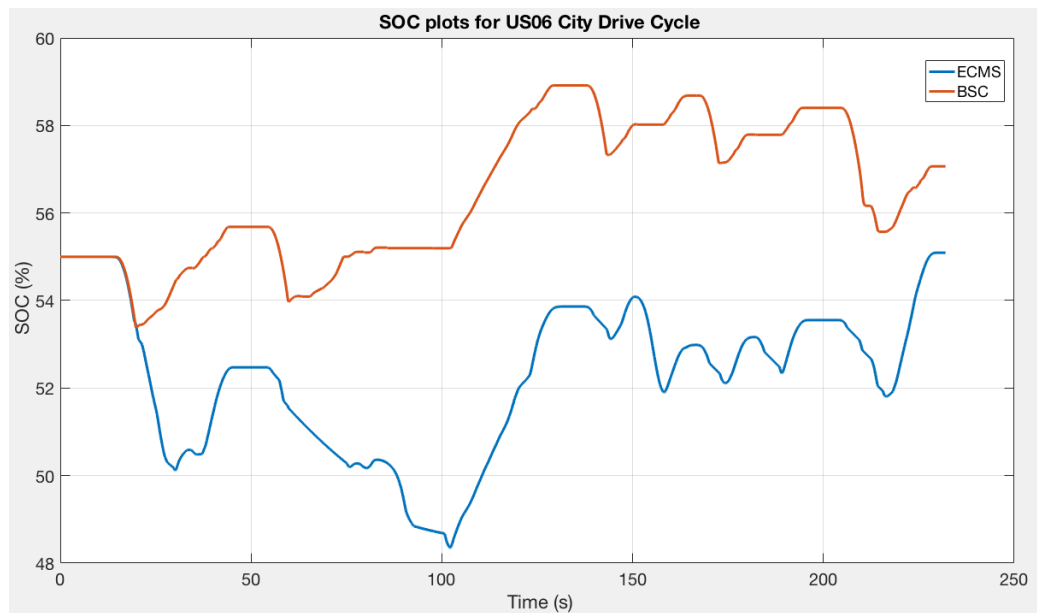


Figure 66: SOC Plots for BSC and ECMS when Simulated for US06 City Drive Cycle

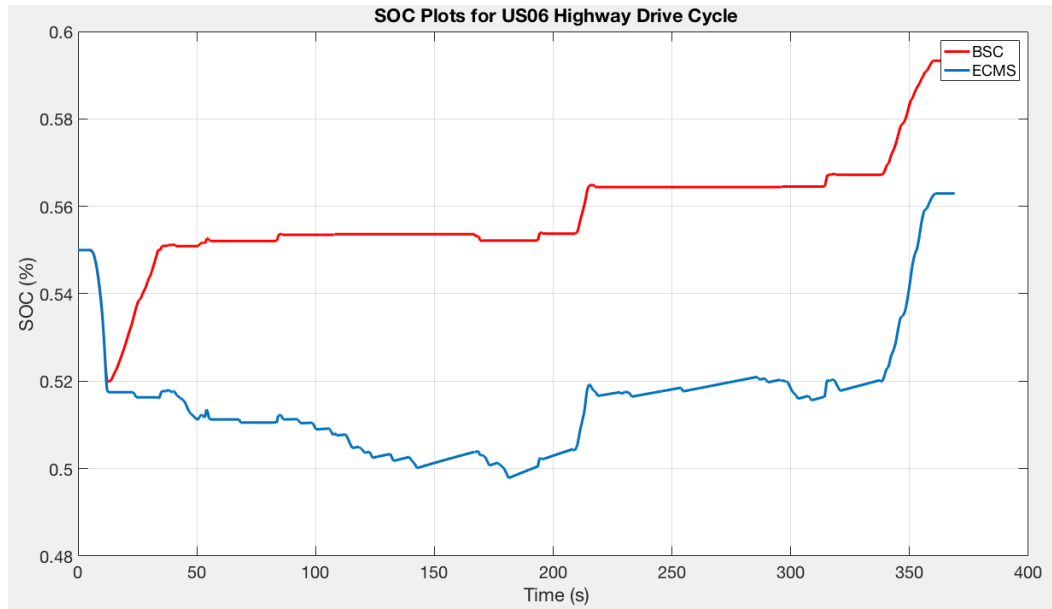


Figure 67: SOC Plots for BSC and ECMS when Simulated for US06 Highway Drive Cycle

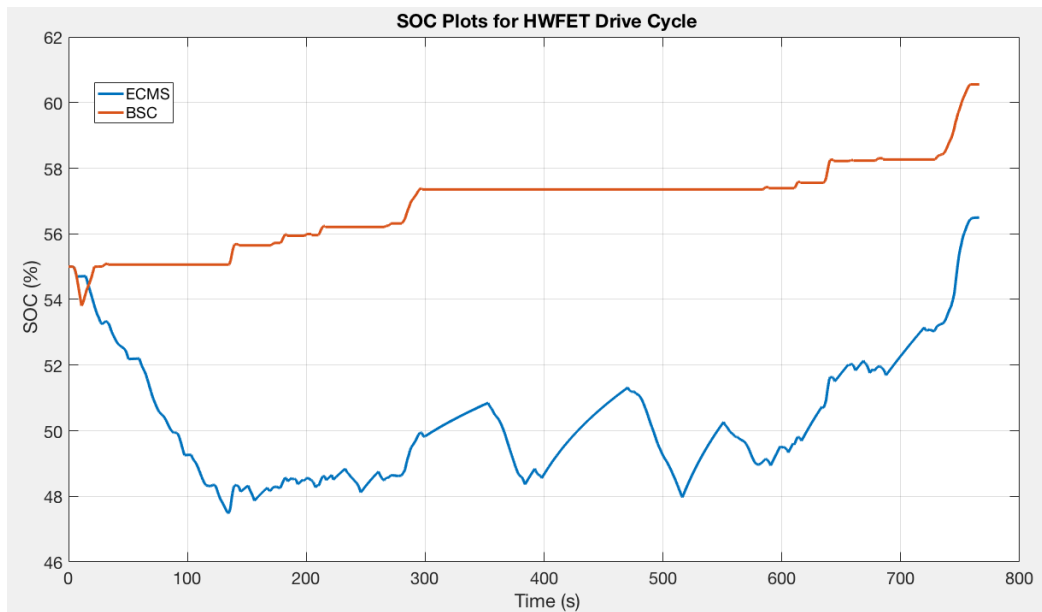


Figure 68: SOC Plots for BSC and ECMS when Simulated for HWFET Drive Cycle



Figure 69: SOC Plots for BSC and ECMS when Simulated for US505 Drive Cycle

The E85 fuel consumed and change in SOC<sup>2</sup> for the Basic Supervisory Controller and ECMS are given in Tables 8 and 9. At a first glance, the difference in raw fuel consumption seems quite large but the change in SOC needs to be accounted for in terms of equivalent fuel. The procedure for converting raw fuel consumption and change in SOC to equivalent fuel consumption is done in Appendix F. This method was developed by Argonne National Labs and is used in the EcoCAR competition to convert raw fuel consumption and SOC change to equivalent miles per gallon in gasoline equivalent. The weighted fuel consumption in (kg/km) of E85 and (L/km) of E85 is given in Tables 10 and 11 respectively. The equivalent gasoline fuel consumption in (L/km) and miles per gallon is given in Tables 12 and 13 respectively. The percentage improvement in fuel economy of the ECMS over the Basic Supervisory Controller (BSC) is given in Table 14.

---

<sup>2</sup>The final SOC in all cases are slightly higher than the initial SOC as the choice of  $s_0$  is higher than the charge sustaining value for each cycle.

Table 8: E85 Fuel Consumption Comparison between Basic Supervisory Controller and ECMS for US06 City and US06 Highway Drive Cycles

	US06 City		US06 Highway	
	Fuel (g)	$\Delta$ SOC(%)	Fuel (g)	$\Delta$ SOC(%)
<b>BSC</b>	325.46	2.16	983.26	4.58
<b>ECMS</b>	286.31	0.18	871.2	1.3

Table 9: E85 Fuel Consumption Comparison between Basic Supervisory Controller and ECMS for HWFET and US505 Drive Cycles

	HWFET		US505	
	Fuel (g)	$\Delta$ SOC(%)	Fuel (g)	$\Delta$ SOC(%)
<b>BSC</b>	1068.4	6.8	575.6	2.3
<b>ECMS</b>	926.1	1.2	501.3	0.08

Table 10: Weighted E85 Fuel Consumption (kg/100km) Comparison for Basic Supervisory Controller and ECMS

	Weighted Fuel Consumption (kg/100km) E85			
	US06 City	US06 Highway	HWFET	US505
<b>BSC</b>	10.43	9.26	5.93	9.44
<b>ECMS</b>	9.96	8.56	5.51	8.65

Table 11: Weighted E85 Fuel Consumption (L/100km) Comparison for Basic Supervisory Controller and ECMS

	<b>Weighted Fuel Consumption (L/100km) E85</b>			
	<b>US06 City</b>	<b>US06 Highway</b>	<b>HWFET</b>	<b>US505</b>
<b>BSC</b>	13.03	11.57	7.42	11.80
<b>ECMS</b>	12.45	10.70	6.91	10.81

Table 12: Equivalent Gasoline Fuel Consumption (kg/100km) Comparison for Basic Supervisory Controller and ECMS

	<b>Equivalent Fuel Consumption (L/100km) Gasoline</b>			
	<b>US06 City</b>	<b>US06 Highway</b>	<b>HWFET</b>	<b>US505</b>
<b>BSC</b>	9.81	8.15	5.22	8.31
<b>ECMS</b>	8.76	7.53	4.86	7.62

Table 13: Fuel Economy (miles/gallon) Gasoline Comparison for Basic Supervisory Controller and ECMS

	<b>Fuel Economy (miles/gallon) Gasoline</b>			
	<b>US06 City</b>	<b>US06 Highway</b>	<b>HWFET</b>	<b>US505</b>
<b>BSC</b>	25.62	28.86	45	28.29
<b>ECMS</b>	26.82	31.21	48.44	30.88

Table 14: Percentage Improvement in Fuel Economy for ECMS over Basic Supervisory Controller

	<b>US06 City</b>	<b>US06 Highway</b>	<b>HWFET</b>	<b>US505</b>
<b>ECMS Improvement</b>	4.69%	8.13%	7.63%	9.13%

It can be seen that the fuel economy for ECMS is better than the Basic Supervisory Controller in all cases. This is due to the fact that the Basic Supervisory Controller was developed based on heuristics and notions of efficient ICE operating regions, whereas in the ECMS, a mathematical model for the fuel consumption that was developed in [16] was used to minimize fuel consumption. This provided definite regions of optimal control modes and explicit equations to calculate the optimal EM power. The Basic Supervisory Controller can be augmented further to achieve a performance close to that of the ECMS but this would involve a great deal of calibration effort and would not be scalable to other powertrain configurations.

### 5.3.2 Consideration of Real World Routes

In the previous subsection, the performance of the ECMS on flat roads was presented. In this section, simulations of the ECMS are done considering elevation changes in real world routes extracted in Chapter 3. The US06 City cycle is simulated for the Stuttgart Route<sup>3</sup>, the US06 Highway cycle is simulated for the Sausalito Route, the HWFET cycle is simulated for the Gatlinburg Route and the US505 cycle is simulated for the Seattle Route. The SOC plots of the simulations for the various drive cycles are shown in Figures 70 to 73.

---

<sup>3</sup>It is worthy to note that the actual distance covered during one US06 City drive cycle is just 2.89 kilometers. The Stuttgart route is 9.1 kilometers long and hence the drive cycle was cascaded three times to simulate US06 City cycle on the Stuttgart route.

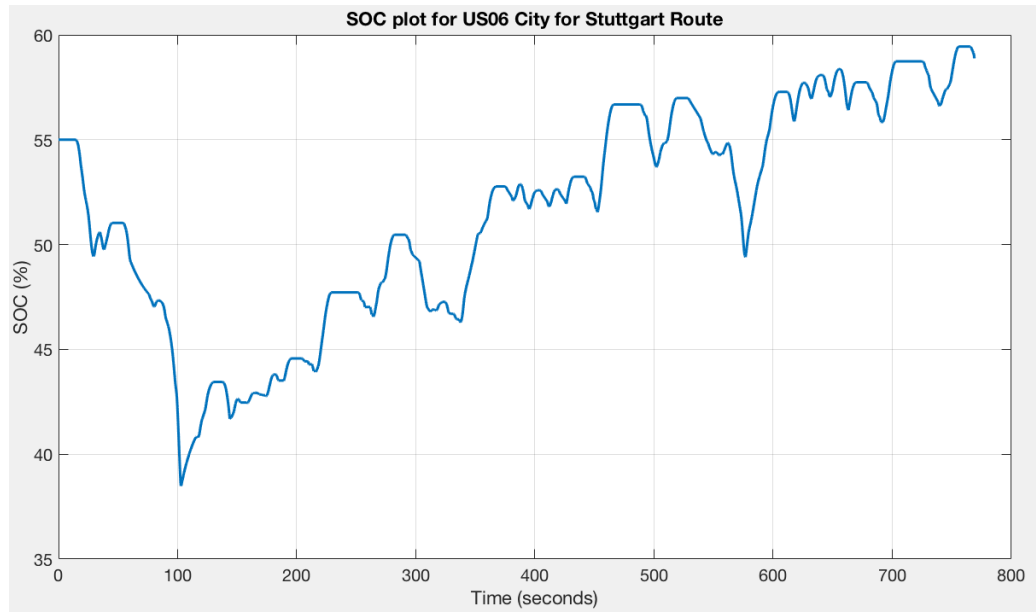


Figure 70: SOC Plot for ECMS Simulation for US06 City Drive Cycle and for Stuttgart Route

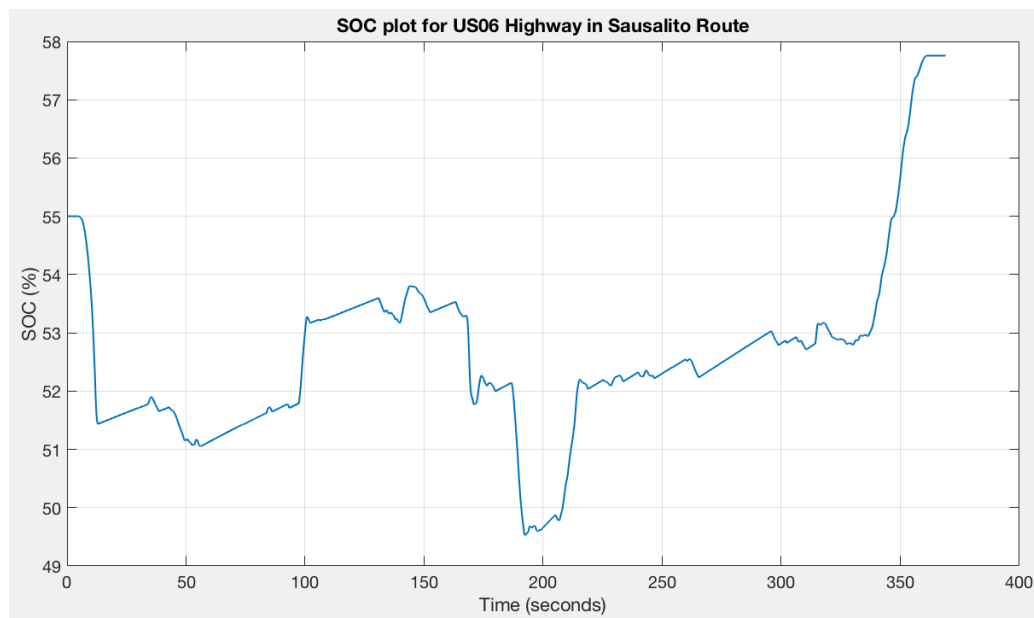


Figure 71: SOC Plot for ECMS Simulation for US06 Highway Drive Cycle and for Sausalito Route

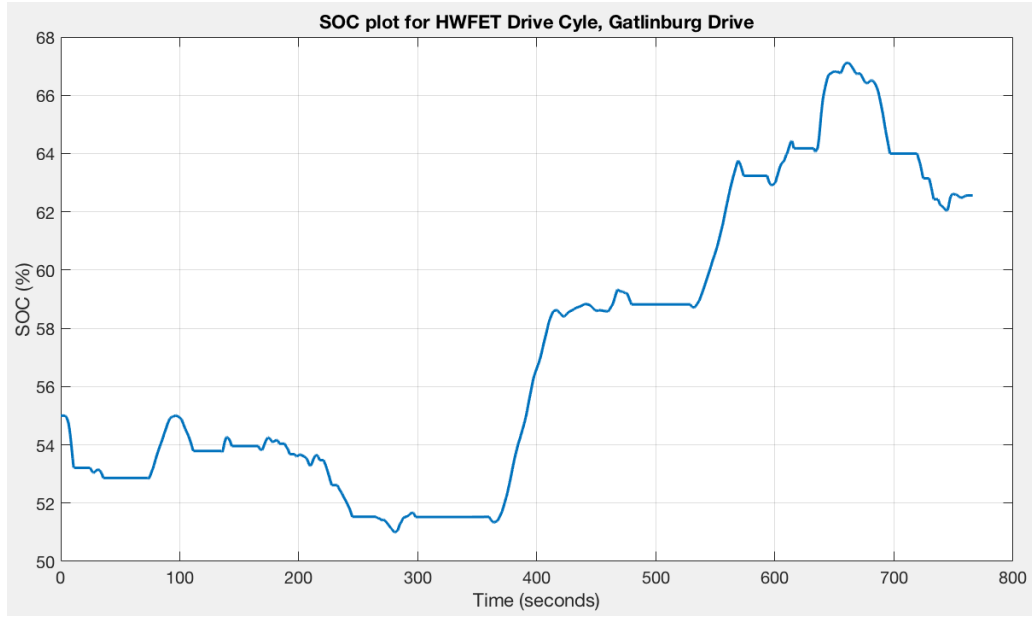


Figure 72: SOC Plot for ECMS Simulation for HWFET Drive Cycle and for Gatlinburg Route

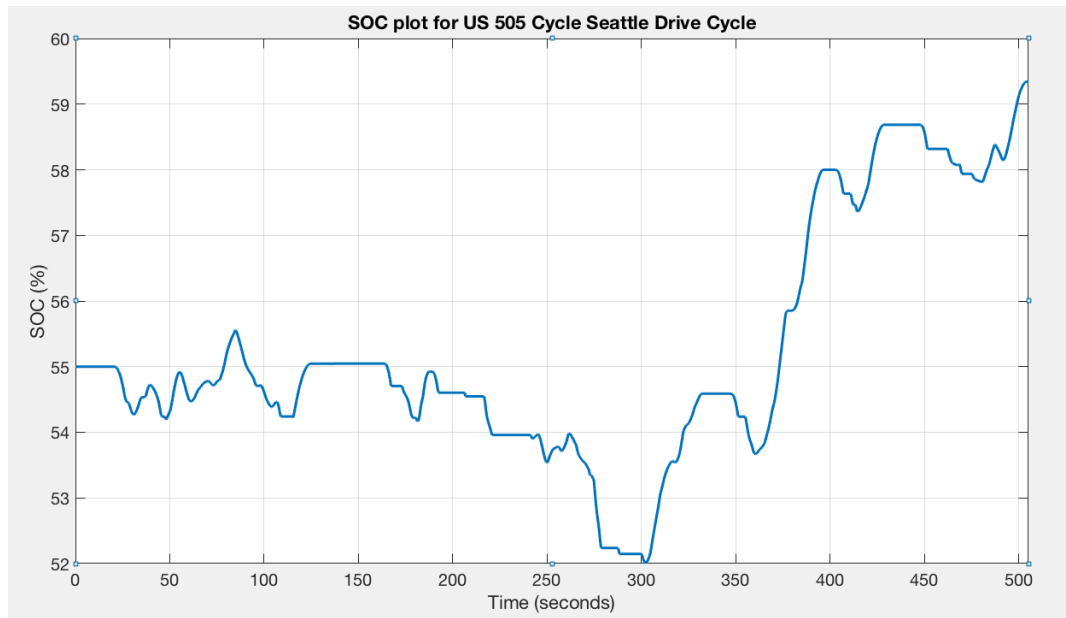


Figure 73: SOC Plot for ECMS Simulation for US505 Drive Cycle and for Seattle Route

The E85 fuel consumed and change in SOC for the ECMS simulations considering



road elevation profiles are given in Tables 15 and 16. The weighted fuel consumption in (kg/km) of E85 and (L/km) of E85 are given in Table 17. The equivalent gasoline fuel consumption in (L/km) and miles per gallon is given in Table 18.

Table 15: E85 Fuel Consumption and Change in SOC for US06 City and US06 Highway Drive Cycles for ECMS

	<b>US06 City</b>		<b>US06 Highway</b>	
	<b>Fuel (g)</b>	<b><math>\Delta</math> SOC(%)</b>	<b>Fuel (g)</b>	<b><math>\Delta</math> SOC(%)</b>
<b>ECMS</b>	1113.9	4	1031.8	3

Table 16: E85 Fuel Consumption and Change in SOC for HWFET and US505 Drive Cycles for ECMS

	<b>HWFET</b>		<b>US505</b>	
	<b>Fuel (g)</b>	<b><math>\Delta</math> SOC(%)</b>	<b>Fuel (g)</b>	<b><math>\Delta</math> SOC(%)</b>
<b>ECMS</b>	1231.2	7.6	506.4	4.34

Table 17: Weighted E85 Fuel Consumption for ECMS

	<b>Weighted Fuel Consumption E85</b>			
	<b>US06 City</b>	<b>US06 Highway</b>	<b>HWFET</b>	<b>US505</b>
<b>(kg/100km)</b>	11.603	9.95	6.858	7.783
<b>(L/100km)</b>	14.097	9.628	6.826	9.235

Table 18: Equivalent Gasoline Fuel Consumption for ECMS

	<b>Equivalent Fuel Consumption Gasoline</b>			
	<b>US06 City</b>	<b>US06 Highway</b>	<b>HWFET</b>	<b>US505</b>
<b>L/100km</b>	10.216	8.760	6.038	6.852
<b>mpg</b>	23.024	26.850	38.951	34.323

These results show that the ECMS achieves good fuel consumption for most flat standard cycles. The performance is affected when elevation profiles is considered. In all simulation cases the final SOC is higher than the reference SOC<sup>4</sup> of 55%. Use of this excess energy during the drive cycle would have reduced fuel consumption in some part of the drive cycle as the electric path is more efficient than the fuel path. If information of these driving conditions were known in advance then the cost associated with using electrical power in the segment leading up to a recuperation phase can be lowered considering that the electrical energy would be regained in the recuperation phase. This can be done only by using predictive strategies [12].

## 5.4 Conclusion

In this chapter an overview of the explicit causal control law derived from ECMS presented in [16] was given. The ECMS strategy was then implemented in Simulink. Simulations were carried out for the US06 City, US06 Highway, HWFET and US505 drive cycles without considering road gradient. The fuel economy of the ECMS was consistently better than the Basic Supervisory Controller in all test cases. Simulations were then carried out considering road gradient for all drive cycles. It was observed

---

<sup>4</sup>This is due to the fact that all the routes selected have a long downward slope following an upward slope. The SOC gradually increases over the downward slope depending on the drive cycle. Only in the case of the HWFET cycle, the SOC falls in the end because the simulation ends on an upward slope.

that consideration of road gradient noticeably changes the performance of the ECMS.

## CHAPTER VI

### PREDICTIVE CONTROL

The ECMS presented in the previous chapter is causal and uses only the present and past states of the powertrain to optimize fuel consumption. This is done by penalizing the use of electric power based on the deviation of the current battery SOC and a constant reference SOC.

Consider a situation where the vehicle is being driven in a hilly region and the vehicle needs to traverse a segment with an upward slope followed by a segment with a downward slope. If the battery SOC is below the set reference SOC, the output of the ECMS would be to traverse the upward slope with a torque split ratio that has a higher ICE torque component. This is due to the fact that the controller has no information about the upcoming downward slope where a possible recuperation would result in the increase of the battery SOC.

If the information about the upcoming segment with a downward slope is known a priori, the corresponding recuperation potential can be used to modify the ECMS to decrease the penalty associated with electric power delivery and further decrease fuel consumption. This would require the controller to be acausal and predictive to a certain degree.

State of the art navigation systems can provide data such as average traveling speeds and elevation profile of upcoming road segments. The control strategy presented in this chapter takes advantage of the availability of this information to improve the performance of the ECMS by generating a time-varying reference SOC trajectory.

This algorithm is called predictive reference signal generator (pRSG) and is presented in [14, 16]. The algorithm uses the elevation profile information for the upcoming road segment, although it can be extended to use information such as road curvature, upcoming intersections etc. As the problem can be formulated as a quadratic program (QP), the underlying algorithm is computationally attractive. Figure 74 shows the extension of ECMS with pRSG [14, 16].

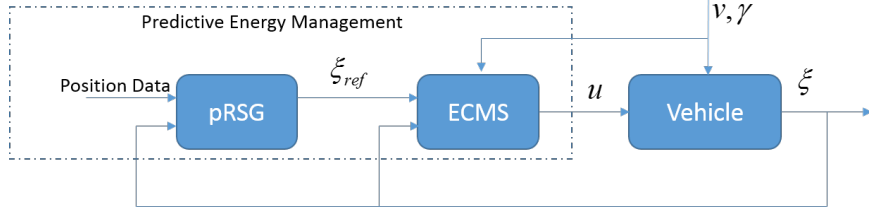


Figure 74: Block Diagram of pRSG - ECMS Algorithm

## 6.1 Method

The operation of an HEV can be classified into two main types. The first type is where the supervisory control can explicitly control the SOC by favoring or penalizing the use of the electrical path. These intervals are referred to as *free* segments. In the case when the powertrain is in recuperation mode, the use of electric power is determined purely by the driver or the drive cycle. These intervals are referred to as *fixed* segments. The prior prediction of these fixed segments is what can be used to further optimize fuel consumption in the preceding free segments. This is achieved by computing a time-varying reference trajectory for the SOC. The prediction of recuperation is done as follows:

- Only the information about the road elevation profile and the average traveling speed provided by the navigation system is used. These data points are specified at the discrete distances.

- The two sets of data are mapped to a common grid. As a result each road segment is characterized by its gradient and its average speed.
- The estimated force acting on the vehicle over each segment is calculated. This force takes into account the air drag, rolling resistance and acceleration due to gravity.
- During some braking events, the power required can exceed the regenerating power capability of the electric path which includes the electric motor, the inverter and the battery. The negative force is thus limited by quotient of electric regenerating power adjusted with the recuperation efficiency, and average velocity.
- The estimated energy recuperated over each segment is calculated as the product of the negative force, the length of the segment and the recuperation efficiency, if the force is negative. The energy recuperated is set to 0 if the force is positive.
- The equivalent SOC change is calculated as the quotient of the recuperated energies and the pack capacity expressed in Joules.
- All successive recuperation segments are lumped into one segment of recuperation. Each of these aggregated recuperation segments is then saved as a triplet containing its future starting time, its duration and the expected SOC change.

The triplet that has the data about the future expected recuperation is used to generate the SOC reference trajectory. A piece-wise affine SOC reference trajectory needs to be computed such that it does not violate the SOC boundaries. This consists of fixed segments that lie between two fixed segments. The starting time, the duration, and the change in SOC, represents the fixed segments. If the starting value of the

SOC for each fixed segment is known, it means that the entire trajectory is defined. Hence, the SOC values are the unknowns and need to be determined in order to fully define the trajectory. The following conditions need to be satisfied by the reference trajectory [14, 16]:

- The reference SOC,  $\xi_{ref}$ , must remain within the interval  $\Omega = [\xi_{ref,min}, \xi_{ref,max}]$  at all times.
- The final SOC of the reference trajectory must remain within the target interval  $\Omega_f = [\xi_{set} - \delta\xi_f, \xi_{set} + \delta\xi_f]$ , where  $\xi_{set}$  is the SOC setpoint for charge sustenance<sup>5</sup>.
- The rate of SOC change during the free elements of the reference trajectory must be minimized.

These constraints can be formulated and solved as a quadratic programming problem as shown in Appendix G. The pRSG is implemented using a Matlab function in Simulink. It uses the vectors of future road gradient and average speed along with the corresponding distance offset vector to produce a reference signal for the battery SOC. The code is given in Appendix H.

The predictive strategy is combined with the non-predictive ECMS strategy as shown in Figure 75. The pRSG generates a position dependent reference SOC signal that is used by the ECMS. Before a recuperation phase, the reference signal is lowered to improve fuel efficiency.

---

<sup>5</sup>For the simulations done in this thesis,  $\xi_{set} = \xi_0 = 0.55$ .

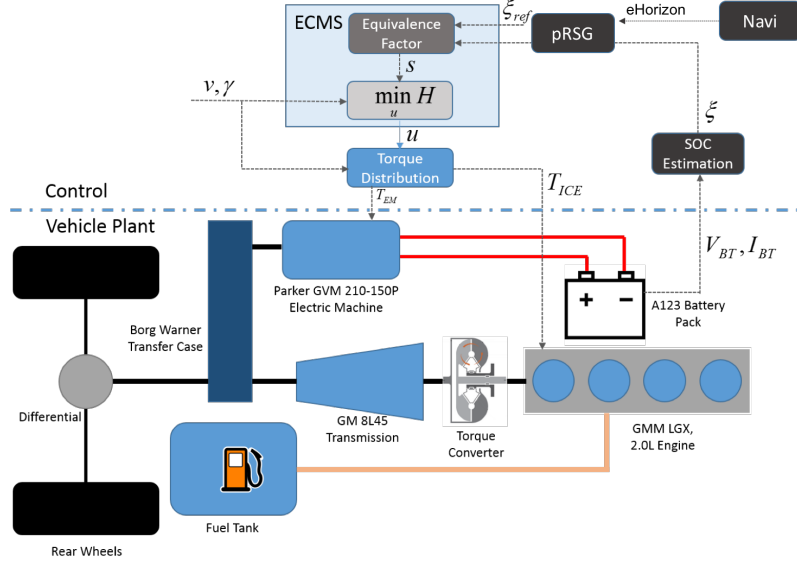


Figure 75: Structure of pRSG-ECMS

## 6.2 Simulation of Navigation Device

To compare the performance of the proposed method, simulations for all three control strategies (Basic Supervisory Controller, ECMS and ECMS-pRSG) must be carried out. The pRSG-ECMS requires the road gradient and average speed data from the navigation system to compute the reference SOC trajectory. The elevation data extracted in Chapter 3 is assumed to be exact and is used for simulation and for prediction. This data is made available to the control strategy as a vector of road gradient points given at 100 meter intervals.

The speed profile cannot be accurately predicted by the navigation device. Using the driving profile prescribed by the standardized driving cycles, an average speed profile for each drive cycle is computed and is used to simulate the output from the navigation system [14, 16]. This approximation is very rough on purpose. It is assumed that the navigation device only recognizes urban, extra-urban and highway driving with corresponding speed limits of 30, 50 and 70 miles per hour respectively.



The average traveling speeds assigned to each of these zones are assumed to be 20, 40 and 60 miles per hour respectively. The algorithm to determine the average driving speed profile from the drive cycle is as follows:

- **Step 1:** The speed signal is converted into an indexed signal  $i(t)$  using threshold values of 35 and 55 miles per hour. Urban, extra-urban and highway driving are represented by indices 1, 2 and 3 respectively.
- **Step 2:** The index signal is filtered using a non-causal zero-phase low-pass filter with a time constant of 30 seconds.
- **Step 3:** The filtered index signal is rounded back to integers.
- **Step 4:** The index signals are converted back to average speeds of 20, 40 or 60 miles per hour.
- **Step 5:** The average speed signal is mapped from a time base to a distance base.

### ***6.3 Simulation Results***

Simulations for the pRSG-ECMS were done for the US06 City, US06 Highway, US505 and HWFET drive cycles considering changes in road gradient.

The plots of the reference SOC<sup>6</sup> generated by the pRSG for the various drive cycles are shown along with the route elevations in Figures 76 to 79. The simulations are done setting the initial and reference SOC to 0.55. The minimum and maximum allowed SOC limits are set to 0.3 and 0.8 respectively. The equivalence factor is computed with parameters given in Table 6.

---

<sup>6</sup>The final value of the reference SOC does not always match the intital SOC due to the choice of values for the softening paramters.

To compare the performance of the pRSG-ECMS with the ECMS and Basic Supervisory Controller, simulations of the ECMS and Basic Supervisory Controller were also carried out with the same conditions as that for the pRSG-ECMS for all the drive cycles.

Plots of the reference SOC generated by the pRSG and the elevation of the road as a function of distance for all routes are shown in Figures 76 to 79.

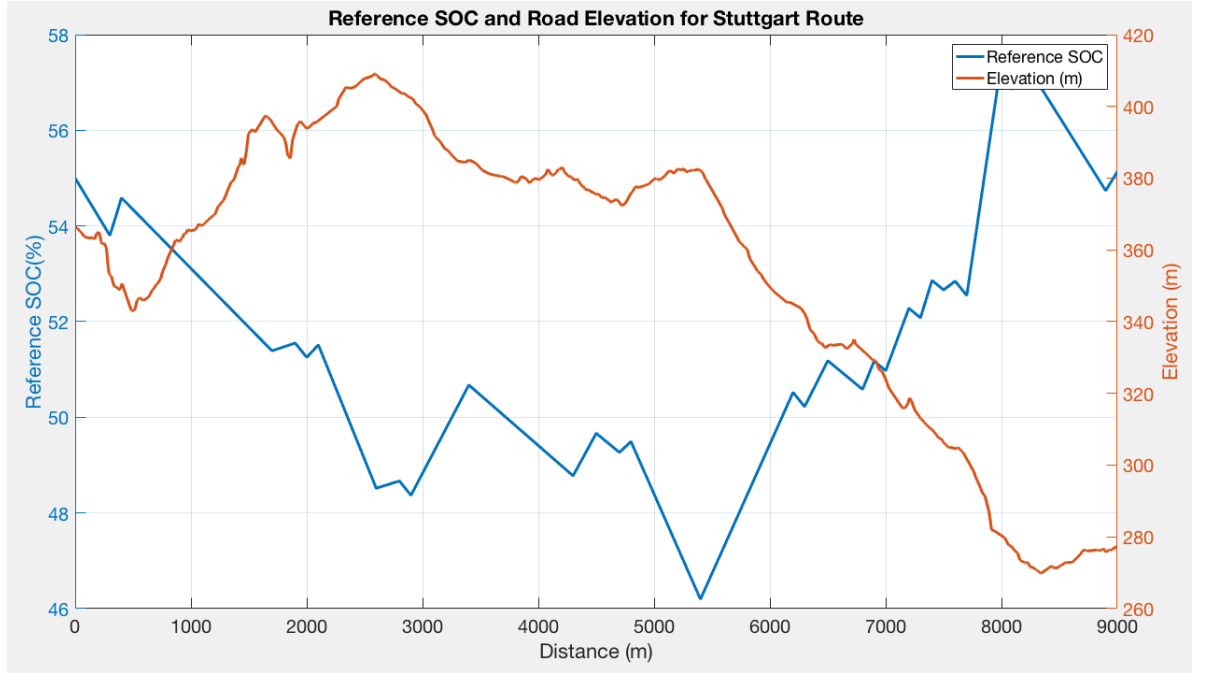


Figure 76: Reference SOC and Road Elevation for Stuttgart Route

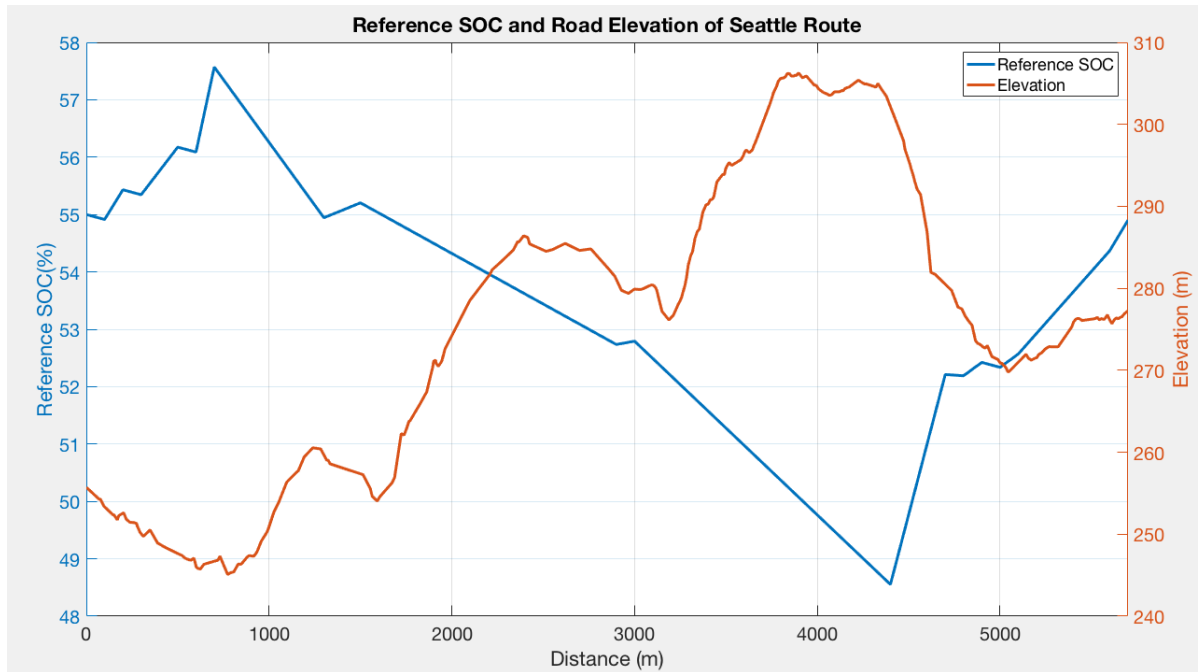


Figure 77: Reference SOC and Road Elevation for Seattle Route

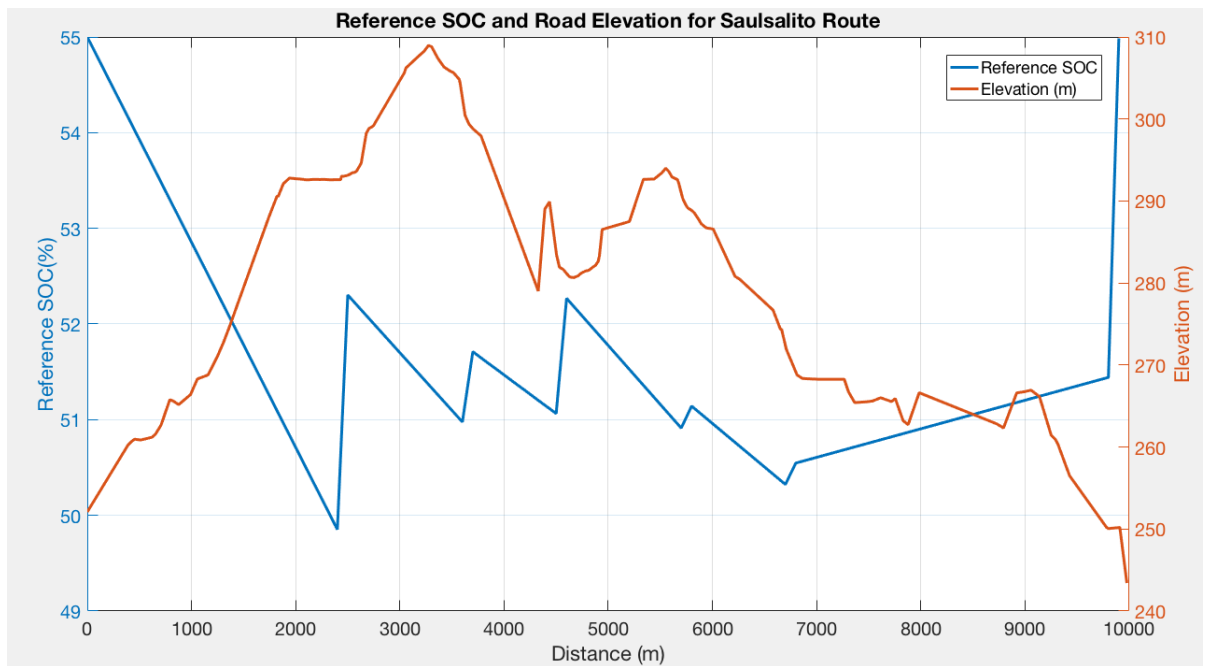


Figure 78: Reference SOC and Road Elevation for Sausalito Route

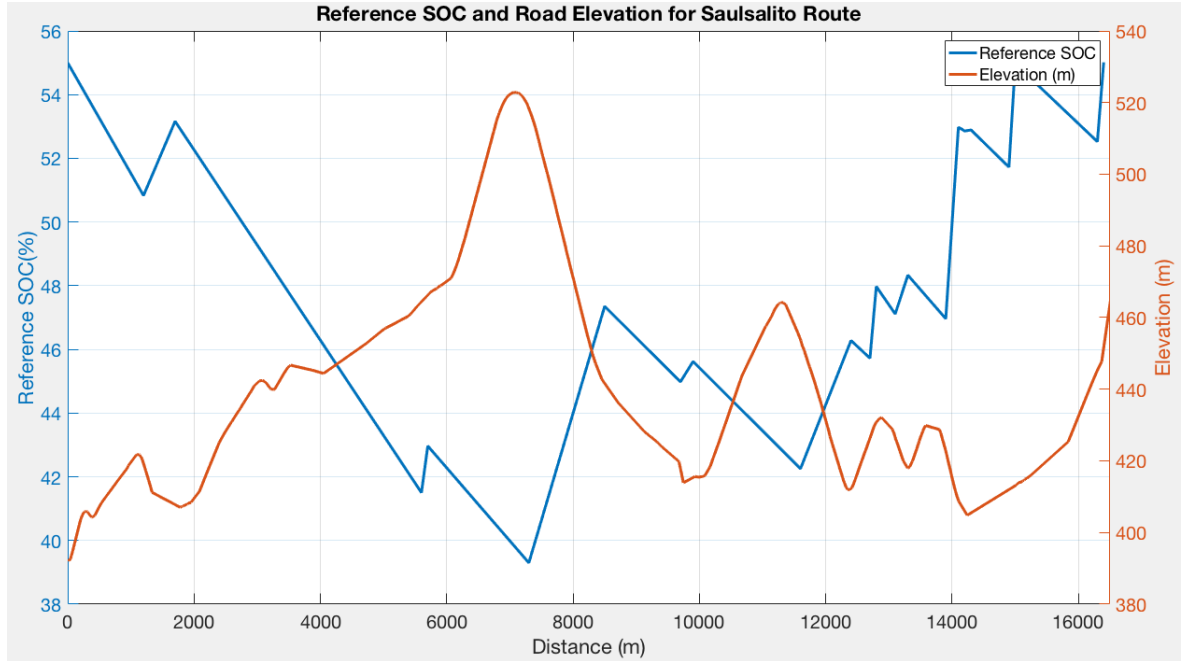


Figure 79: Reference SOC and Road Elevation for Gatlinburg Route

A comparison of the SOC plots for the Basic Supervisory Controller, the ECMS and the pRSG-based ECMS is shown in Figures 80 to 83.

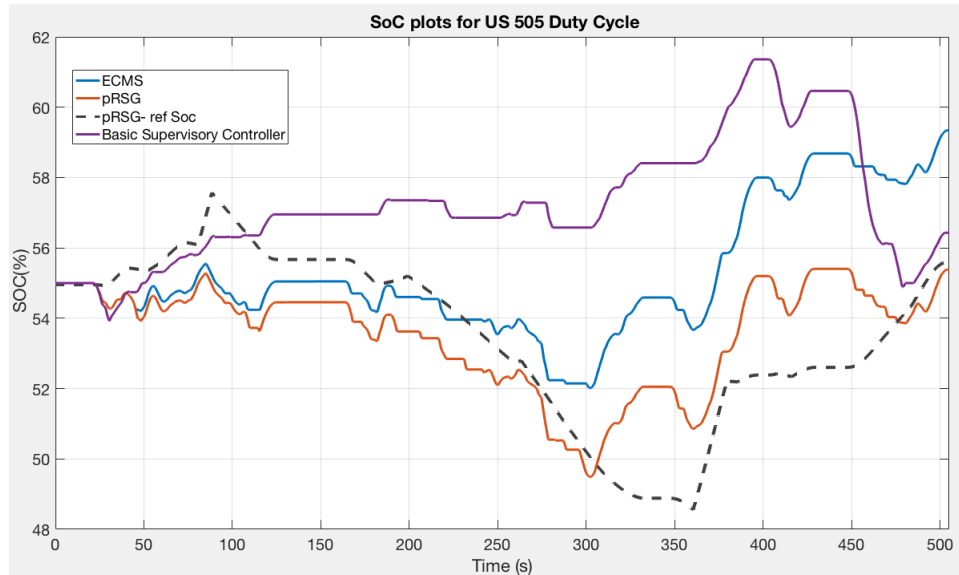


Figure 80: SOC Plots for Simulation of US505 Cycle in Seattle Route

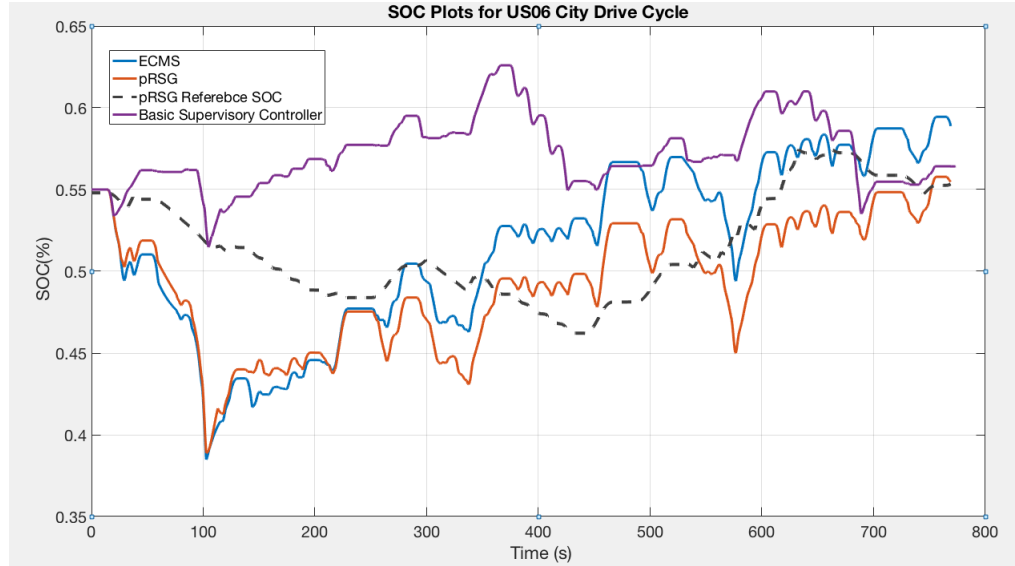


Figure 81: SOC Plots for Simulation of US06 City Cycle in Stuttgart Route

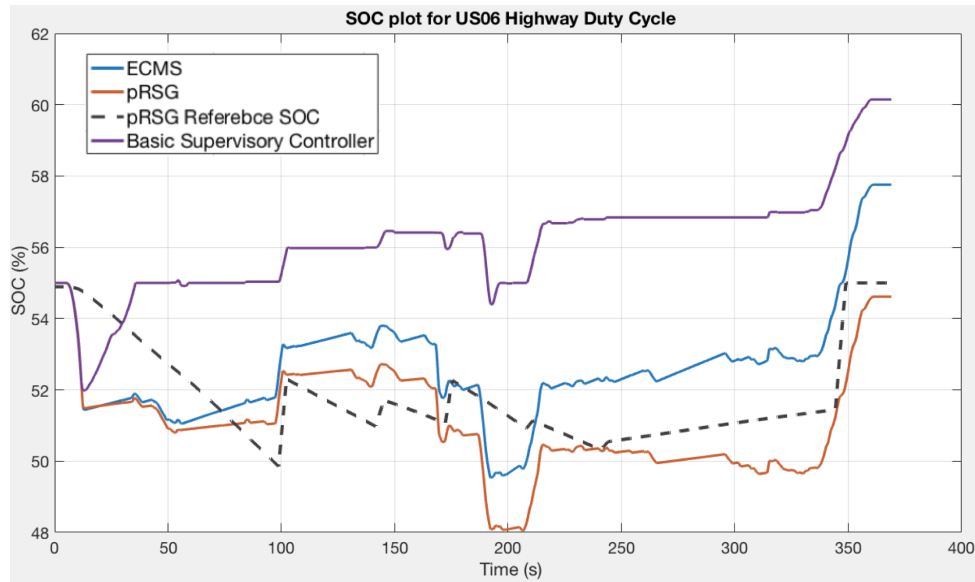


Figure 82: SOC Plots for Simulation of US06 Highway Cycle in Sausalito Route

Table 19: E85 Fuel Consumption Comparison for US06 City and US06 Highway Drive Cycles for Basic Supervisory Controller, ECMS and pRSG

	US06 City		US06 Highway	
	Fuel (g)	$\Delta$ SOC(%)	Fuel (g)	$\Delta$ SOC(%)
<b>BSC</b>	1472.8	1.2	1156.2	5
<b>ECMS</b>	1113.9	4	1031.8	3
<b>pRSG-ECMS</b>	1024.4	0.5	959.41	-0.4

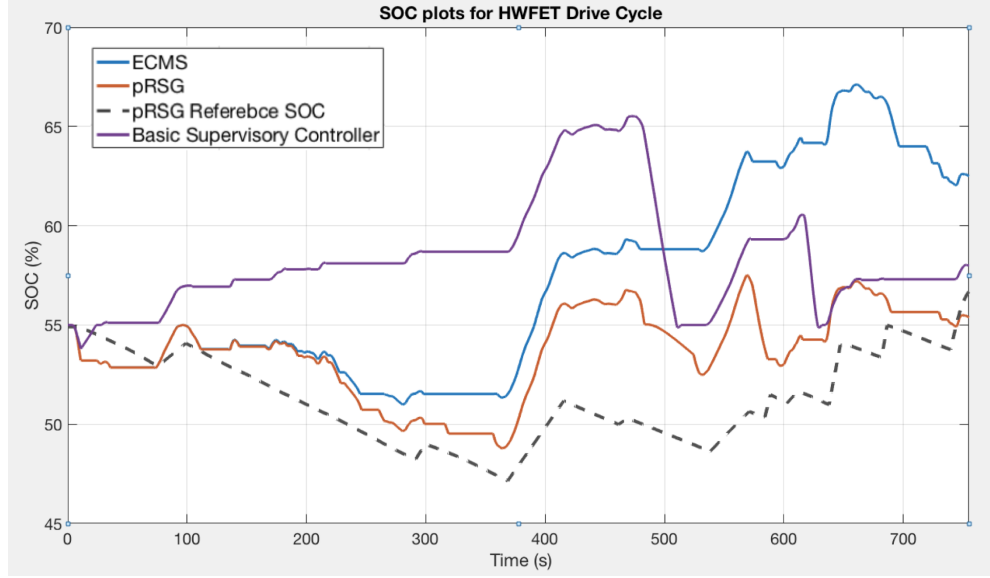


Figure 83: SOC Plots for Simulation of HWFET Cycle in Gatlinburg Route

The E85 fuel consumed and the change in SOC for the Basic Supervisory Controller, ECMS and pRSG-ECMS is given in Tables 19 and 20. The weighted fuel consumption of E85 in (kg/km) and (L/km) is given in Tables 21 and 22. The equivalent gasoline fuel consumption in (L/km) and miles per gallon<sup>7</sup> is given in Tables 23 and 24. The percentage improvement in fuel economy of the pRSG-ECMS over ECMS is given in Table 25.

The pRSG generates a reference SOC signal based on the average traveling speed and

<sup>7</sup>The mileage of the 2016 Camaro SS with a 3.6 liter engine and 8-speed gearbox is reported to be 19 miles per gallon in the city and 28 miles per gallon in the highway.

Table 20: E85 Fuel Consumption and SOC Change Comparison for HWFET and US505 Drive Cycles for Basic Supervisory Controller, ECMS and pRSG

	<b>HWFET</b>		<b>US505</b>	
	<b>Fuel (g)</b>	<b><math>\Delta</math> SOC(%)</b>	<b>Fuel (g)</b>	<b><math>\Delta</math> SOC(%)</b>
<b>BSC</b>	1319.5	3	624.4	1.4
<b>ECMS</b>	1231.2	7.6	506.4	4.34
<b>pRSG-ECMS</b>	1106.5	0.4	441.5	0.3

Table 21: Weighted E85 Fuel Consumption (kg/100km) Comparison for Basic Supervisory Controller, ECMS and pRSG

	<b>Weighted Fuel Consumption (kg/100km) E85</b>			
	<b>US06 City</b>	<b>US06 Highway</b>	<b>HWFET</b>	<b>US505</b>
<b>BSC</b>	16.012	10.935	7.753	10.489
<b>ECMS</b>	11.603	9.95	6.858	7.783
<b>pRSG-ECMS</b>	11.185	9.668	6.671	7.549

Table 22: Weighted E85 Fuel Consumption (L/100km) Comparison for Basic Supervisory Controller, ECMS and pRSG

	<b>Weighted Fuel Consumption (L/100km) E85</b>			
	<b>US06 City</b>	<b>US06 Highway</b>	<b>HWFET</b>	<b>US505</b>
<b>BSC</b>	20.016	13.669	9.69	13.11
<b>ECMS</b>	14.504	12.437	8.573	9.729
<b>pRSG-ECMS</b>	13.982	12.085	8.339	9.436

Table 23: Equivalent Gasoline Fuel Consumption (L/100km) Comparison for Basic Supervisory Controller, ECMS and pRSG

	<b>Equivalent Fuel Consumption (L/100km) Gasoline</b>			
	<b>US06 City</b>	<b>US06 Highway</b>	<b>HWFET</b>	<b>US505</b>
<b>BSC</b>	14.097	9.628	6.826	9.235
<b>ECMS</b>	10.216	8.760	6.038	6.852
<b>pRSG-ECMS</b>	9.847	8.512	5.873	6.646

Table 24: Fuel Economy (miles/gallon) Gasoline Comparison for Basic Supervisory Controller, ECMS and pRSG

	<b>Fuel Economy (miles/gallon) Gasoline</b>			
	<b>US06 City</b>	<b>US06 Highway</b>	<b>HWFET</b>	<b>US505</b>
<b>BSC</b>	16.684	24.43	34.456	25.469
<b>ECMS</b>	23.024	26.850	38.951	34.323
<b>pRSG-ECMS</b>	23.884	27.632	40.045	35.389

Table 25: Percentage Improvement in Fuel Economy for pRSG-ECMS over ECMS

	<b>US06 City</b>	<b>US06 Highway</b>	<b>HWFET</b>	<b>US505</b>
<b>pRSG Improvement</b>	3.73%	2.91%	2.81%	3.1%

the road gradient. In each case the reference SOC is lowered in the region before an upcoming downward slope or braking event. This reduces the cost associated with using electric power. This allows higher power delivery from the EM during this segment, thus reducing fuel consumption. The battery SOC curve does not follow the reference SOC trajectory accurately as only average traveling speed is used to make the prediction and also because vehicle stop events cannot be predicted by the navigation device. It can be seen that in all cases, the pRSG-ECMS shows a consistent improvement in fuel efficiency over the non-predictive ECMS.

## 6.4 *Conclusion*

The non-predictive control strategy developed in the previous chapter is extended with the pRSG algorithm that generates a reference SOC trajectory based on future road profiles. The reference trajectory developed is computationally effective and relies on data only from the navigation device which makes it applicable on a real vehicle. Simulation results have shown that the fuel consumption can be reduced when compared to the non-predictive strategy for driving cycles with topographic profiles.



## CHAPTER VII

### SUMMARY AND CONCLUSIONS

The aim of this thesis was to develop control strategies for the Georgia Tech EcoCAR3 P3 parallel hybrid electric vehicle. The vehicle model was developed in Simulink and a Basic Supervisory Controller based on heuristics was first developed to understand the working of a parallel hybrid electric vehicle. To improve the fuel efficiency of the vehicle, an optimal control strategy based on Pontryagin's Minimum Principle was developed. A causal, nonpredictive controller called *equivalent consumption minimization strategy* was used. The algorithm minimizes an equivalent fuel cost calculated as the sum of the fuel consumed by the engine and the weighted electric energy consumed from the battery. A simplified plant model was used to obtain an explicit control law [16]. The control law is a simple rule-based algorithm dependent only on vehicle parameters and is computationally inexpensive.

The ECMS achieved a fuel consumption that was better than the Basic Supervisory Controller. When the road gradient changes were considered, the performance of the ECMS changed slightly. Given that there is an upcoming recuperation phase, the fuel consumption in the ECMS can be reduced further by lowering the cost associated with the use of electric power in the segment leading up to the recuperation phase. This can only be achieved if some information regarding the upcoming road segments is available.

This thesis uses an algorithm called predictive reference signal generator (pRSG) developed in [16]. This algorithm uses data such as road gradient and average speed

from a navigation for a planned trip and produces a reference state of charge trajectory that the controller tries to follow till the end of the trip. The pRSG and ECMS together form a predictive energy management system. As a result, good performance in terms of fuel consumption was achieved even in driving cycles with changing elevation profiles.

## APPENDIX A

### TRANSMISSION CONTROL MODULE

The GM 8-speed 8L45 transmission has a dedicated transmission control module that controls the gear shifts. The shifts are decided based on up-shift and down-shift maps that are flashed into the controller. The up-shift map prescribes a vehicle speed threshold for each gear position and accelerator pedal position. If the vehicle speed exceeds this threshold, the TCM initiates an up-shift action. Similarly the down-shift map prescribes a lower vehicle speed threshold. If the vehicle speed is below this threshold, the TCM initiates a down-shift action. These curves are shown in Figure 84.

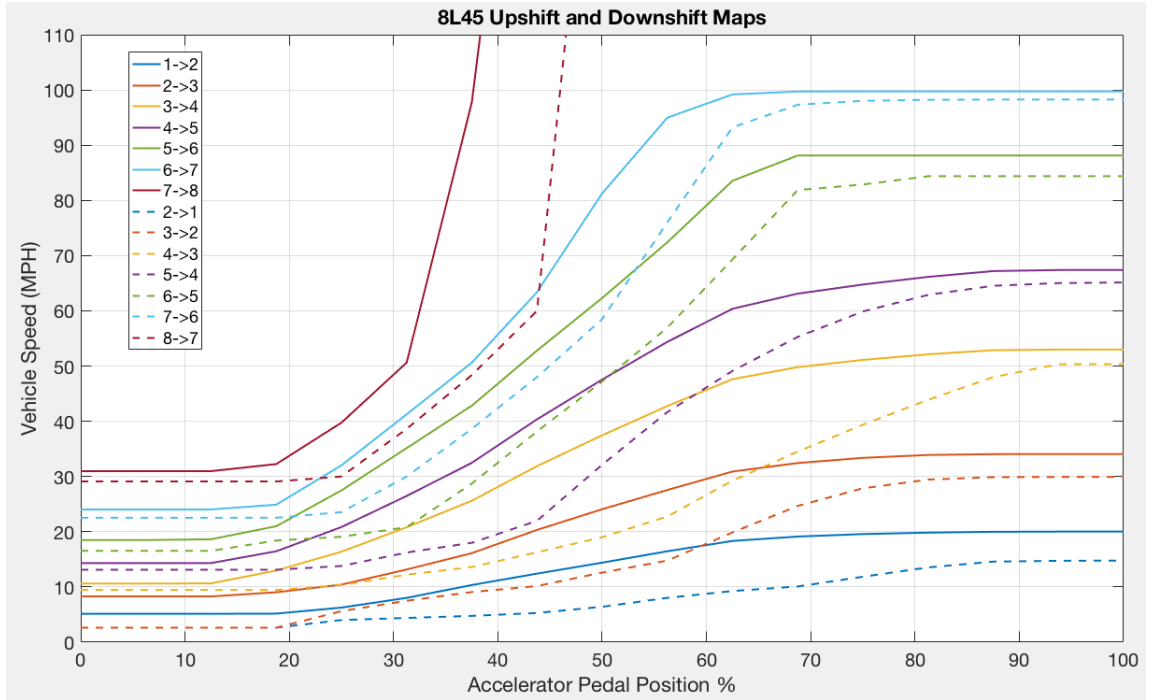


Figure 84: Transmission Up-shift and Down-shift Threshold Curves

In the vehicle model, the TCM is implemented as a state machine. This is shown in Figure 85.

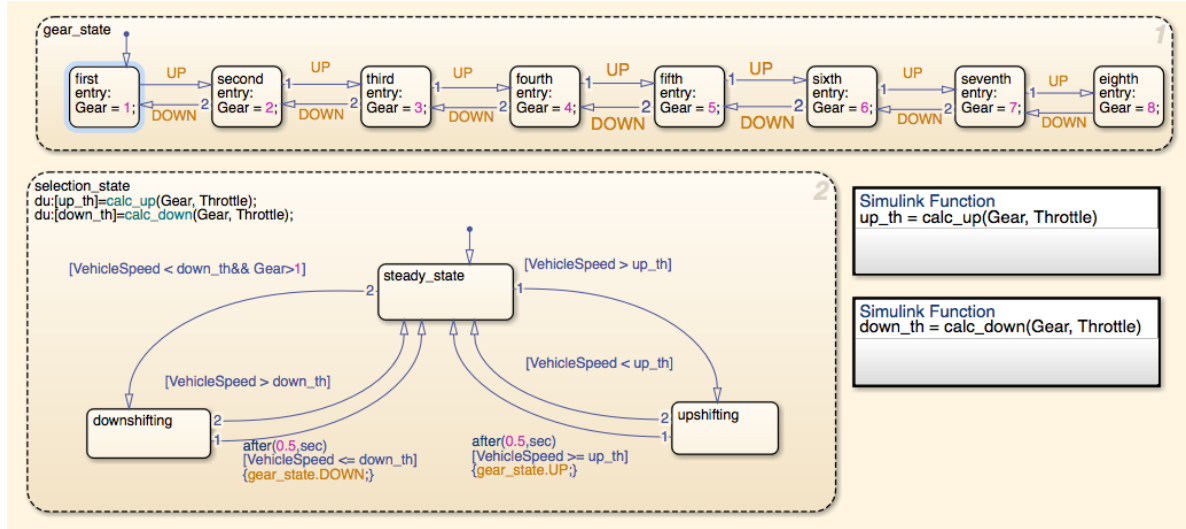


Figure 85: TCM State Machine

This implementation uses two parallel state machines. The first one called *Gear State*, has 8 states that represent each gear position. Each of these states simply outputs the corresponding gear number that is used by the transmission plant model to engage the right gear. There are two transition events that are defined: *UP* and *DOWN*, that are initiated by the other state machine, *Selection State*. When *UP* is initiated, the gear state moves up by one. Similarly the gear state moves down by one when *DOWN* is initiated.

The state machine *Selection State* uses vehicle speed and engine throttle request<sup>8</sup> as input to calculate the upshift and downshift thresholds for the current gear position using the two Simulink functions. The state machine has three states: *Steady State*, *Upshifting* and *Downshifting*. When the control is in *Steady State* and the

<sup>8</sup>In a conventional vehicle, the transmission shift maps use the accelerator pedal position as the input. In essence this is also the engine throttle request since no other power source exists. In the hybrid vehicle considered, since the EM is coupled after the transmission, the engine throttle request is used as the input for the shift maps and not accelerator pedal position.

vehicle speed is greater than the calculated upshift threshold, the control moves to *Upshifting* state. If the condition that the vehicle speed is greater than the calculated upshift threshold persists for 0.5 seconds, the *UP* transition event for *Gear State* is initiated and the control returns to *Steady State*. If the vehicle speed is greater than the upshift threshold when in *Upshifting* state, the control returns to *Steady State* without any action. Similarly, if the vehicle speed is less than the downshift threshold, the control moves to *Downshifting*. If this condition persists for more than 0.5 seconds, the *DOWN* action is initiated and the control moves back to *Steady State*. If the vehicle speed exceeds the downshift threshold when in *Downshifting* state, the control moves back to *Steady State* without any action.

This implementation of the TCM and the gear shift strategy is used for the Basic Supervisory Controller, ECMS and the pRSG-ECMS strategies.

## APPENDIX B

### ENGINE CONTROL MODULE

In this thesis the engine control module is modeled such that it controls starting and stalling action of the engine. It also sets an engine throttle request based on the normalized torque request commanded by the supervisory controller and the engine state. This is implemented as a state machine using a Stateflow chart in Simulink as shown in Figure 86.

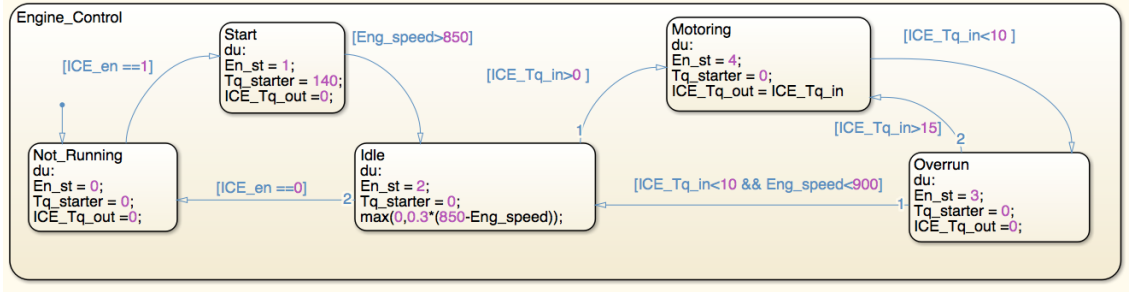


Figure 86: Engine Control State Machine

The state machine uses normalized ICE torque request and the ICE enable state  $ICE\_en$  (from the supervisory controller) and the ICE speed as inputs. It produces the  $ICE\_Tq\_out$ , the throttle request set for the throttle body and starter motor torque request  $Tq\_starter$  as outputs. The ground state is called *Not Running*. In this state, the ICE throttle request and the starter motor torque request are set to 0 and the ICE is turned off. When an ICE enable request is set by the supervisory controller, the control moves to *Start* mode where the starter motor torque request is set to a positive value to start the ICE. Once the ICE speed reaches idling speed (850 RPM in this case), the control moves to *Idle* mode, where the starter motor torque

request is set to 0 and the ICE throttle request is set using a proportional controller to maintain idling speed. If the normalized ICE torque request is greater than 0, the control moves to *Motoring* mode. Here the ICE throttle request is set equal to the normalized torque request commanded by the supervisory controller. When operating in *Motoring* mode, if the normalized ICE torque request is 0, the control moves to *Overrun* mode where the ICE throttle request is set to 0 and the engine is dragged by the powertrain. When operating in *Overrun* mode, if the normalized ICE torque request is 0 and the engine speed is less than 900 RPM, the control moves to *Idling* mode. Else when operating in *Overrun* mode, if the normalized ICE torque request is greater than 0, the control moves back to *Motoring* mode. When in idling mode, if the ICE enable flag is set to 0, the control moves to *Not Running* state.

## APPENDIX C

### DRIVE CYCLES

A plot of the US06 City drive cycle is shown in Figure 87.

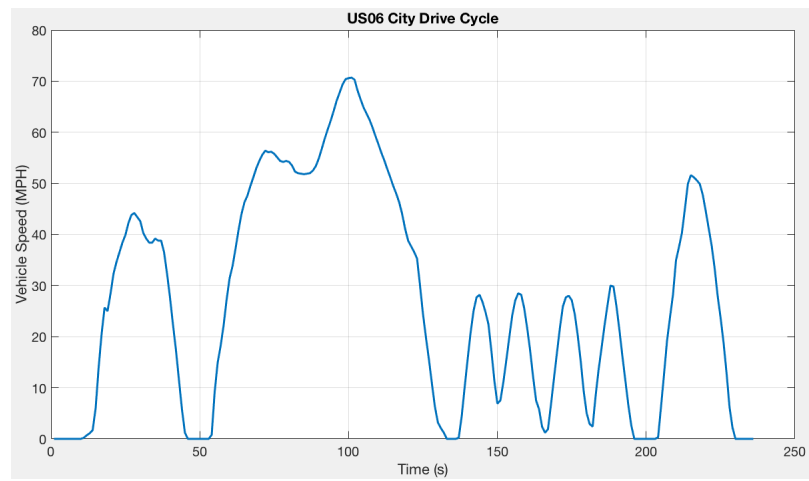


Figure 87: US06 City Drive Cycle

A plot of the US06 Highway drive cycle is shown in Figure 88.



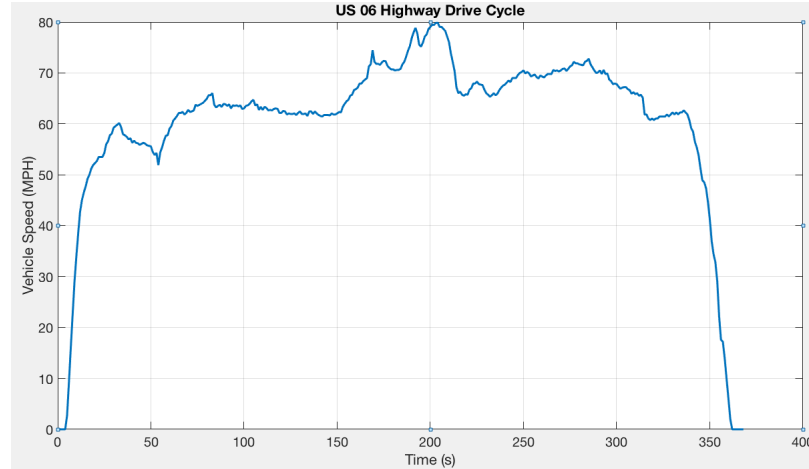


Figure 88: US06 Highway Drive Cycle

A plot of the HWFET drive cycle is shown in Figure 89.

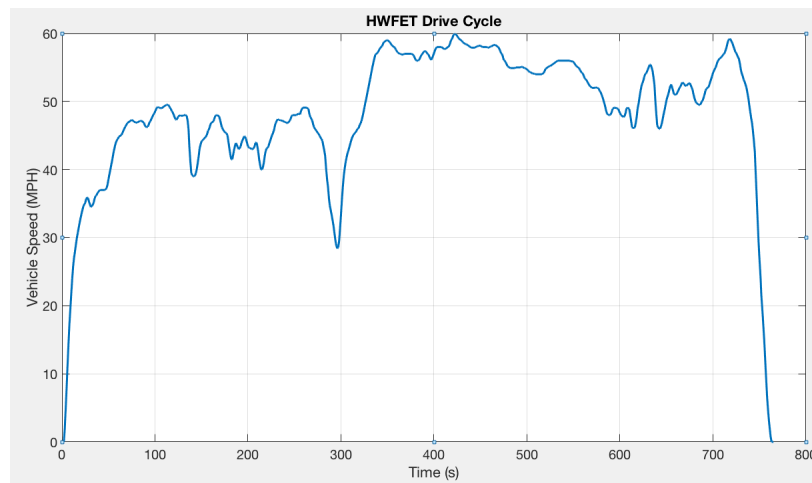


Figure 89: HWFET DriveCycle

A plot of the US505 drive cycle is shown in Figure 90.

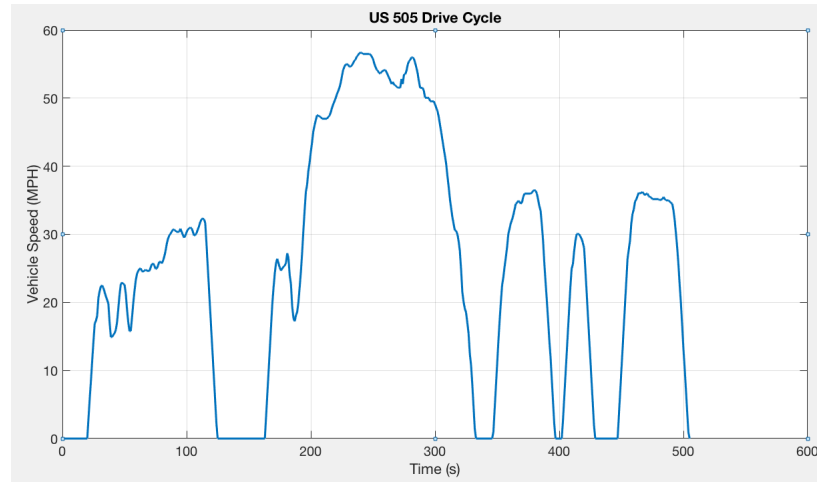


Figure 90: US505 Drive Cycle

## APPENDIX D

### EQUIVALENCE FACTOR AND POWER THRESHOLD CALCULATIONS FOR ECMS

The optimal EM power for the simplified plant model when not considering the EM and ICE power limits is given by [16]:

$$P_{EM}^o(P_d, s) = \begin{cases} \frac{\eta\left(\frac{\eta}{e}-s\right)}{2s\beta} & \text{if } (P_d > 0) \text{ and } (s < \frac{\eta}{e}) \text{ and } (P_d > P_{lim}^{bo}(s)) \\ 0 & \text{if } (P_d > 0) \text{ and } (\frac{\eta}{e} < s < \frac{1}{\eta e}) \text{ and } (P_d > P_{lim}^{th}(s)) \\ -\frac{s-\frac{1}{\eta e}}{2s\beta\eta} & \text{if } (P_d > 0) \text{ and } (s > \frac{1}{\eta e}) \text{ and } (P_d > P_{lim}^{re}(s)) \\ -\frac{s-\frac{1}{\eta e}}{2s\beta\eta} & \text{if } (P_d < 0) \text{ and } (P_d > P_{lim}^{re-}) \\ P_d & \text{otherwise} \end{cases} \quad (41)$$

When considering the limits of the EM to be  $P_{EM_{Max}}$  and  $P_{EM_{Min}}$ , and ICE power limit to be  $P_{ICE_{Max}}$ , the optimal EM power candidates are given by:

$$\mathcal{P}(P_d, s) = \{0, P_d, P_{EM}^{bo}(s), P_{EM}^{re}(s), P_{EM}^{Max}(s), P_{EM}^{Min}(s), P_d - P_{ICE}^{Max}(s)\} \quad (42)$$

where  $P_{EM}^o$  is the optimal EM power,  $P_d$  is the driver power demand,  $\eta$  is the EM efficiency,  $e$  is the ICE efficiency and  $\beta$  is the electric path efficiency. When applying the component power limits to Equation 41, the explicit optimal control law is given

by:

$$P_{EM}^o(P_d, s) = \begin{cases} \frac{\eta\left(\frac{\eta}{e} - s\right)}{2s\beta} & \text{if } (P_d > 0) \text{ and } (s < \frac{\eta}{e}) \text{ and } (P_d > P_{lim}^{bo}) \\ P_d - P_{ICEMax} & \text{if } (P_d > P_{ICEMax} + P_{EM}^{bo}) \text{ and } (s < \frac{\eta}{e}) \\ 0 & \text{if } (P_d > 0) \text{ and } (\frac{\eta}{e} < s < \frac{1}{\eta e}) \text{ and } (P_d > P_{ICEMax}) \\ P_d - P_{ICEMax} & \text{if } (\frac{\eta}{e} < s < \frac{1}{\eta e}) \text{ and } (P_d > P_{lim}^{th}) \\ -\frac{s - \frac{1}{\eta e}}{2s\beta\eta} & \text{if } (P_d > 0) \text{ and } (s > \frac{1}{\eta e}) \text{ and } (P_d > P_{lim}^{re}) \\ P_d - P_{ICEMax} & \text{if } (s > \frac{1}{\eta e}) \text{ and } (P_d > P_{ICEMax} + P_{EM}^{re}) \\ -\frac{s - \frac{1}{\eta e}}{2s\beta\eta} & \text{if } (P_d < 0) \text{ and } (P_d > P_{lim}^{re-}) \\ P_d & \text{otherwise} \end{cases} \quad (43)$$

$$s_{Boost} = \frac{\eta}{e} \quad (44)$$

$$s_{Gen} = \frac{1}{\eta e} \quad (45)$$

$$P_{EM}^{re} = -\frac{s - \frac{1}{\eta e}}{2s\beta\eta} \quad (46)$$

$$P_{EM}^{bo} = \frac{\eta\left(\frac{\eta}{e} - s\right)}{2s\beta\eta} \quad (47)$$

$$P_{lim}^{bo} = \frac{-\left(\frac{s}{\eta} - \frac{1}{e}\right) + \sqrt{4\frac{s\beta P_0}{\eta^2 e}}}{2\frac{s\beta}{\eta^2}} \quad (48)$$

$$P_{lim}^{th} = \frac{-\left(\frac{s}{\eta} - \frac{1}{e}\right) + \sqrt{\left(\frac{s}{\eta} - \frac{1}{e}\right)^2 + 4\frac{s\beta P_0}{\eta^2 e}}}{2\frac{s\beta}{\eta^2}} \quad (49)$$

$$P_{lim}^{re} = \frac{-\left(\frac{s}{\eta} - \frac{1}{e}\right) + \sqrt{\left(\frac{s}{\eta} - \frac{1}{e}\right)^2 - \left(\frac{s}{\eta} - \frac{1}{e\eta^2}\right)^2 + 4\frac{s\beta P_0}{\eta^2 e}}}{2\frac{s\beta}{\eta^2}} \quad (50)$$

$$P_{lim}^{re-} = \frac{-\left(s\eta - \frac{1}{e}\right) + \sqrt{4\frac{s\beta\eta^2 P_0}{e}}}{2s\beta\eta^2} \quad (51)$$

where  $P_0$  is the engine friction power.

## APPENDIX E

### SIMULINK IMPLEMENTATION FOR ECMS POWER LIMIT CALCULATIONS

This appendix shows the Simulink implementation for the power limits used in ECMS for mode transitions. The power threshold values for mode transitions  $P_{EM}^{re}$ ,  $P_{EM}^{bo}$ ,  $P_{lim}^{bo}$ ,  $P_{lim}^{th}$  and  $P_{lim}^{re}$  are done according to the equations in Appendix D and the corresponding Simulink implementations are shown in Figures 91 to 96.

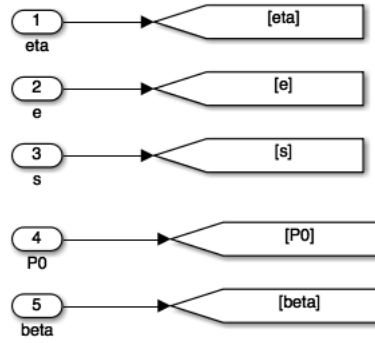


Figure 91: ECMS Power Limit - Inputs

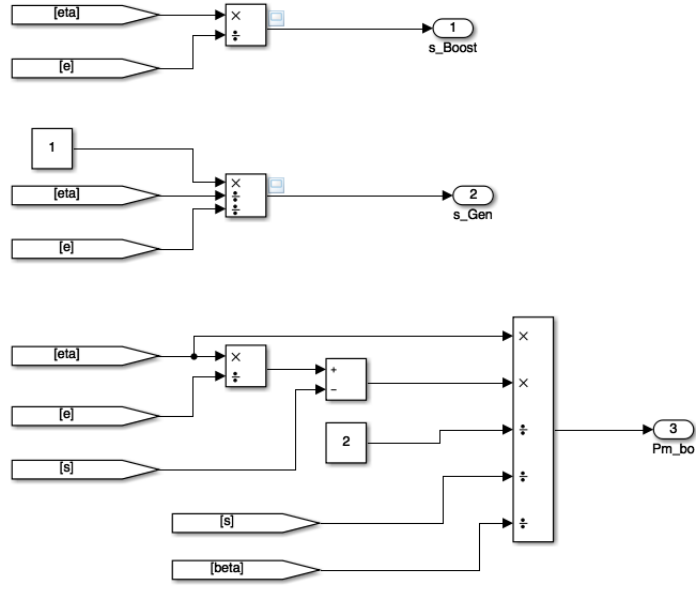


Figure 92: ECMS Power Limit Calculations -  $s_{Boost}$ ,  $s_{Gen}$  and  $P_{EM}^{bo}$

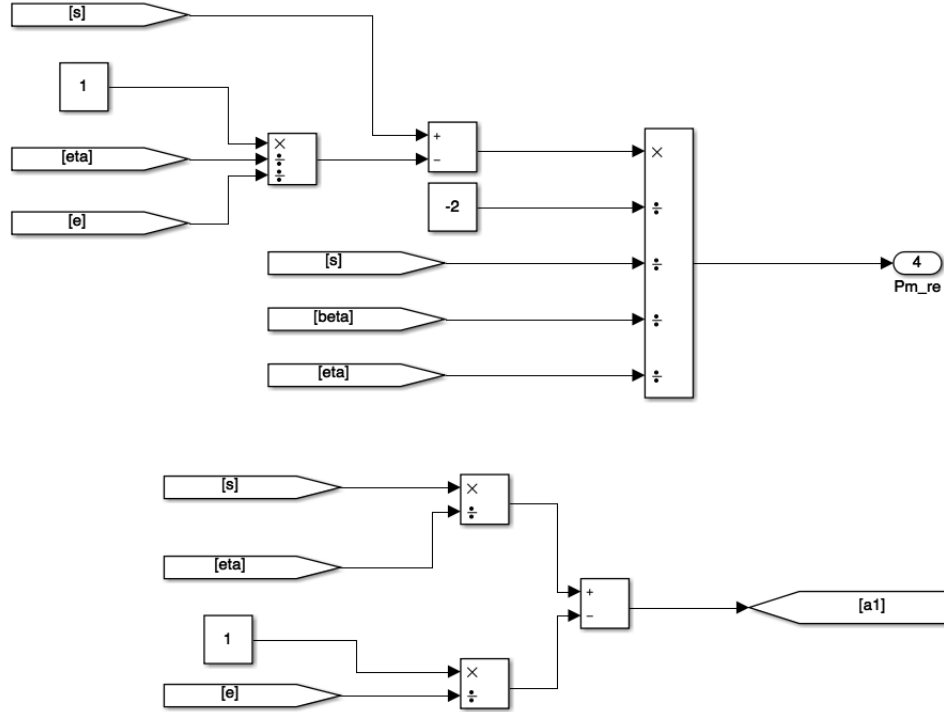


Figure 93: ECMS Power Limit Calculations -  $P_{EM}^{re}$  and  $\frac{s}{\eta} - \frac{1}{e}$

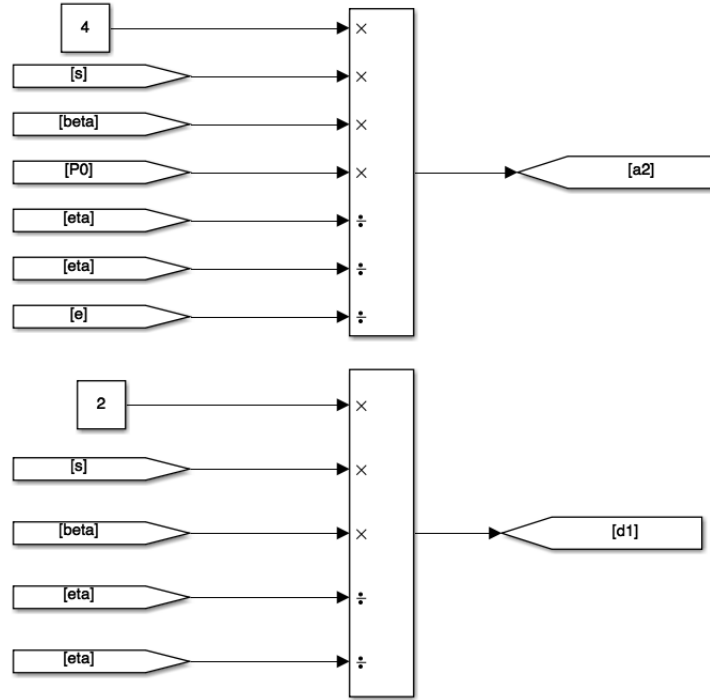


Figure 94: ECMS Power Limit Calculations -  $\frac{4s\beta P_0}{\eta^2 e}$  and  $\frac{2s\beta}{\eta^2}$

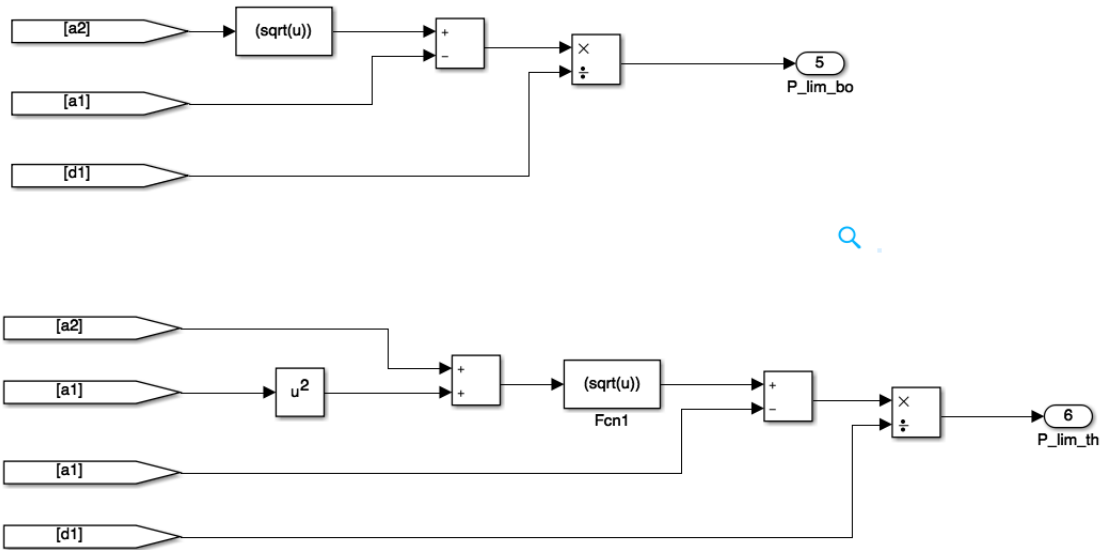


Figure 95: ECMS Power Limit Calculations -  $P_{lim}^{bo}$  and  $P_{lim}^{th}$

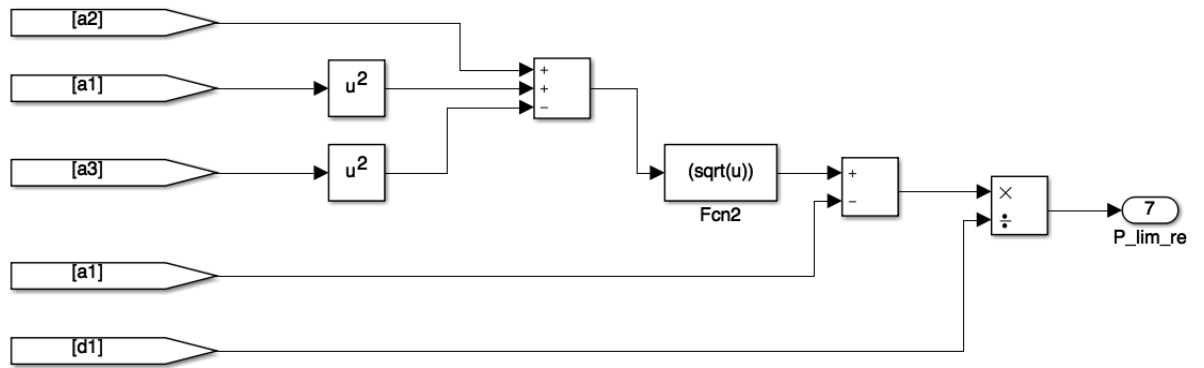


Figure 96: ECMS Power Limit Calculations -  $P_{lim}^{re}$



## APPENDIX F

### CALCULATION OF EQUIVALENT GASOLINE FUEL CONSUMPTION

The raw fuel consumption and the net change in SOC between initial and final conditions are derived for each simulation. The comparison of the performance of each control algorithm cannot be done by just looking at the raw fuel consumption. A correction of equivalent fuel corresponding to the net change in SOC needs to be done first. The SOC corrected fuel consumption in kg/km is calculated as:

$$FC_{SOC-Corrected} = \frac{Mass_{fuel} + \frac{EC_{electric}}{0.25 \cdot LHV_{fuel}}}{Cycle_{distance}} \quad (52)$$

where  $Mass_{fuel}$  is the mass of fuel consumed in kg,  $EC_{electric}$  is the electric consumption in kWh,  $LHV_{fuel}$  is the lower heating value of the fuel, and  $Cycle_{distance}$  is the distance traveled in the drive cycle expressed in km. The lower heating value for E85 is 7.96 kWh/kg. Given that the battery pack used for the simulation is rated at 8Ah at 324V, the energy consumption in kWh for  $\Delta\xi$  net change in SOC is given by:

$$EC_{electric} = \frac{\Delta\xi \times 324 \times 8 \times 3600}{3.6 \times 10^6} = 2.5848 \times \Delta\xi \quad (53)$$

The fuel consumption in L/100km is given by:

$$FC_{E85} = \frac{FC_{SOC-Corrected} \times 100}{\rho_{E85}} \quad (54)$$

where  $\rho_{E85}$  is the density of E85. This value is taken to be 0.8 kg/L. The fuel consumption in L/100km gasoline equivalent is given by:

$$FC_{gas} = FC_{E85} \cdot \frac{FED_{E85}}{FED_{gas}} \quad (55)$$

where  $FED_{E85}$  and  $FED_{gas}$  are the fuel energy density values in kWh/L for E85 and gasoline respectively. The values for  $FED_{E85}$  and  $FED_{gas}$  are 6.265 kWh/L and 8.895 kWh/L respectively.  $FC_{gas}$  is converted to miles per gallon of gasoline equivalent (mpgge) as:

$$FC_{mpgge} = \frac{0.621371}{100 \times 0.264172 \times FC_{gas}} \quad (56)$$

## APPENDIX G

### PRSG PROBLEM FORMULATION

The reference trajectory  $\xi_{ref}$  generated by the pRSG must:

- remain within the interval  $\Omega = [\xi_{ref,min}, \xi_{ref,max}]$  at all times.
- keep the final SoC of the reference trajectory within a target interval  $\Omega = [\xi_{set} - \delta\xi_f, \xi_{set} + \delta\xi_f]$
- minimize the rate of change of the SoC during the free elements of the reference trajectory. These constraints can be expressed mathematically as follows:

$$\tilde{\xi}_i \geq \xi_{ref,min} - \varepsilon_i \quad (57a)$$

$$\tilde{\xi}_i + \Delta\tilde{\xi}_i \leq \xi_{ref,max} + \varepsilon_i \quad (57b)$$

$$\tilde{\xi}_N + \Delta\tilde{\xi}_N \geq \xi_{set} - \delta\xi_f - \varepsilon_f \quad (57c)$$

$$\tilde{\xi}_N + \Delta\tilde{\xi}_N \leq \xi_{set} + \delta\xi_f + \varepsilon_f \quad (57d)$$

In the above equations,  $\varepsilon_i$  and  $\varepsilon_f$  are slackness variables used to guarantee the feasibility by softening the state constraints. Equation 57a has to be fulfilled for all  $i = 1, \dots, N$ . The condition in Equation 57b has to be fulfilled only for  $i = 1, \dots, N - 1$ . For  $i = N$ , the condition given by Equation 57b is always fulfilled when the condition given by Equation 57d is fulfilled. This gives a total of  $2N + 1$  inequalities that need to be fulfilled.

The free segments and the penalties on slackness variables are minimized as:

$$\min_z \left\{ \alpha_{\varepsilon,f} \cdot \varepsilon_f^2 + \sum_{i=1}^N \left( \frac{\Delta\hat{\xi}_i^2}{\Delta\hat{\ell}_i^2} + \alpha_{\varepsilon} \cdot \varepsilon_i^2 \right) \right\} \quad (58a)$$

Table 26: Parameters for the pRSG-ECMS

Parameter	Value	Unit
$\eta_{recup}$	0.8	-
$\xi_{min}$	0.3	-
$\xi_{max}$	0.8	-
$\delta\xi_f$	0.02	-
$\alpha_\varepsilon$	$10^6$	$s^{-2}$
$\alpha_{\varepsilon,f}$	$10^5$	$s^{-2}$

where

$$\Delta\hat{\xi}_i = \tilde{\xi}_i - (\tilde{\xi}_{i-1} + \Delta\tilde{\xi}_{i-1}) \quad (58b)$$

$$\Delta\tilde{\xi}_0 = 0 \quad (58c)$$

$$\Delta\hat{t}_i = \tilde{t}_i - (\tilde{t}_{i-1} + \Delta\tilde{t}_{i-1}) \quad (58d)$$

$$\tilde{t}_0 = 0 \quad (58e)$$

$$\Delta\tilde{t}_0 = 0 \quad (58f)$$

The superscript  $\hat{\cdot}$  represents variables of the free segments. The unknown vector  $z$  of length  $2N + 1$  is given by:

$$z = [\tilde{\xi}_1, \dots, \tilde{\xi}_N, \varepsilon_1, \dots, \varepsilon_N, \varepsilon_f]^T, \quad (59)$$

where  $\tilde{\xi}_i$  represent the SOC at the beginning of each  $i$ -th segment. The slackness variables are penalized using the positive weights  $\alpha_\varepsilon$  and  $\alpha_{\varepsilon,f}$ . The parameters for the pRSG-ECMS is given in Table 26.

The problem can be formulated as a standard quadratic program [14, 16]:

$$\min_z z^T Q z + p^T z \quad (60a)$$

s.t.

$$Az \leq b \quad (60b)$$

where

$$Q_{i,j} = \begin{cases} \frac{1}{\Delta t_i^2} + \frac{1}{\Delta t_{i+1}^2}, & \text{if } i = j = 1, \dots, N-1 \\ \frac{1}{\Delta t_N^2}, & \text{if } i = j = N \\ \alpha_\varepsilon, & \text{if } i = j = N+1, \dots, 2N \\ \alpha_{\varepsilon,f}, & \text{if } i = j = 2N+1 \\ -\frac{1}{\Delta t_i^2}, & \text{if } i = j-1 = 2, \dots, N \\ -\frac{1}{\Delta t_i^2}, & \text{if } i = j+1 = 1, \dots, N-1 \\ 0, & \text{otherwise} \end{cases} \quad (61)$$

$$p_i = \begin{cases} -\frac{2\xi_0}{\Delta t_N^2}, & \text{if } i = N \text{ and } N = 1 \\ -\frac{2\xi_0}{\Delta t_1^2} + \frac{2\Delta\tilde{\xi}_1}{\Delta t_2^2}, & \text{if } i = 1 \text{ and } N > 1 \\ -\frac{2\Delta\tilde{\xi}_{i-1}}{\Delta t_i^2} + \frac{2\Delta\tilde{\xi}_i}{\Delta t_{i+1}^2}, & \text{if } i = 2, \dots, N-1 \text{ and } N > 2 \\ -\frac{2\Delta\tilde{\xi}_{N-1}}{\Delta t_N^2}, & \text{if } i = N \text{ and } N > 1 \\ 0, & \text{if } i = N+1, \dots, 2N+1 \end{cases} \quad (62)$$

$$A_{i,j} = \begin{cases} -1, & \text{if } i = 1, \dots, 2N-1 \text{ and } j = i \\ -1, & \text{if } i = 1, \dots, N \text{ and } j = i+N \\ 1, & \text{if } i = N+1, \dots, 2N-1 \text{ and } j = i-N \\ -1, & \text{if } i = 2N \text{ and } j = N \\ -1, & \text{if } i = 2N \text{ and } j = 2N+1 \\ 1, & \text{if } i = 2N+1 \text{ and } j = N \\ -1, & \text{if } i = 2N+1 \text{ and } j = 2N+1 \\ 0, & \text{otherwise} \end{cases} \quad (63)$$

$$b_i = \begin{cases} -\xi_{ref,min}, & \text{if } i = 1, \dots, N \\ \xi_{ref,max} - \Delta\tilde{\xi}_{i-N}, & \text{if } i = N+1, \dots, 2N-1 \\ -\xi_{set} + \delta\xi_f + \Delta\tilde{\xi}_N, & \text{if } i = 2N \\ \xi_{set} + \delta\xi_f - \Delta\tilde{\xi}_N, & \text{if } i = 2N+1 \end{cases} \quad (64)$$

## APPENDIX H

### MATLAB CODE FOR THE PRSG

```
function SOC_setpt = ...
    pRSG_func(dt_offset, avg_spd, gamma_in, len, veh_offset)

persistent d SOC_ref offset_mem flag val_flag SOC_set_mem;

if isempty(d)
    d = (-100:100:1000000);
    SOC_ref = 0.55*ones(size(d));
    SOC_set_mem = 0.55;
    offset_mem = -1;
    flag = 1<0;
    val_flag = 0>1;
end

if(len>1)
    dist_offset = dt_offset(1:len);
    avg_speed = avg_spd(1:len);
    gamma = gamma_in(1:len);

    if(dist_offset(1) ≠ offset_mem)
        offset_mem = dist_offset(1);
        flag = 1>0;
    end
end
```

```

if flag
    rho = 1.225;
    Af = 2.2220;
    cd = 0.3440;
    mv = 1809;
    g=9.81;
    seg_len = 100;

    alpha = -0.4736;
    beta = 1.0316;
    A_coeff = 0.0684;
    B_coeff = 1.0316e-4;
    C_coeff = 4.1258e-7;
    P= 35; %psi;

    NF = mv*g*cos(gamma);
    cr = P^alpha.*(A_coeff+B_coeff*avg_speed + ...
        C_coeff*avg_speed.^2).*sign(avg_speed);
    F = mv*g*sin(gamma)+0.5*rho*Af*cd*avg_speed.^2 + ...
        cr.*NF.^beta ;

    idx = find(F<0);
    E = zeros(size(F));
    E(idx) = -F(idx)*seg_len*0.7;

    dSOC = E/(6.2*3600*324);
    dt_tilde_avg = seg_len./avg_speed;
    t_tilde_avg = zeros(size(dt_tilde_avg));
    for i = 1:length(dt_tilde_avg)-1
        t_tilde_avg(i+1) = t_tilde_avg(i)+dt_tilde_avg(i);
    end

```

```

end

idx = find(E>0);
t_tilde = t_tilde_avg(idx);
dt_tilde = dt_tilde_avg(idx);

dist_tilde = dt_offset(idx);
dSOC_max = 0.01*dt_tilde;
dSOC_tilde = min(dSOC(idx),dSOC_max);

idx = 1;

x1 = zeros(100,1);
x2 = zeros(100,1);
x3 = zeros(100,1);
x4 = zeros(100,1);
x5 = zeros(100,1);

x1(idx)=0;
x2(idx)=0;
x3(idx)=t_tilde(1);
x4(idx)=dist_tilde(1);
x5(idx)=0;
for i = 1:length(dSOC_tilde)
    x1(idx) = x1(idx) + dSOC_tilde(i);
    x2(idx) = x2(idx) + dt_tilde(i);
    x5(idx) = x5(idx) + seg_len;
    if (i<length(dSOC_tilde))
        if(t_tilde(i+1)≠t_tilde(i)+dt_tilde(i))
            idx = idx+1;
            x3(idx)=t_tilde(1+i);
            x4(idx)=dist_tilde(1+i);
            x1(idx)=0;

```



```

        x2(idx)=0;
        x5(idx)=0;
    end
end
end

dSOC_tilde = x1(1:idx);
dt_tilde = x2(1:idx);
t_tilde = x3(1:idx);
dist_tilde = x4(1:idx);
del_dist_tilde = x5(1:idx);

SOC_min = 0.3
SOC_max = 0.8
SOC_set =0.55;
dSOC_f = 0.02;
alpha = 10^6;
alpha_f = 10^5;

N = idx;
if(N>1)
    dt_hat = zeros(N,1);
    for i = 1:N
        if i >1
            dt_hat(i) = t_tilde(i) - (t_tilde(i-1) + ...
                dt_tilde(i-1));
        else
            dt_hat(i) = t_tilde(i);
        end
    end
end

Q = zeros(2*N+1);

```

```

A = zeros(2*N+1);
b = zeros(2*N+1,1);
p = zeros(2*N+1,1);

for i = 1:2*N+1
    for j = 1:2*N+1
        if (i==j)
            if (i<N-1)
                Q(i,j) = 1/dt_hat(i)^2 + 1/dt_hat(i+1)^2;
            elseif (i==N)
                Q(i,j) = 1/dt_hat(i)^2;
            elseif (i>N)&&(i≤2*N)
                Q(i,j) = alpha;
            elseif (i==2*N+1)
                Q(i,j) = alpha_f;
            end
        elseif(i == j-1)&&(i<N)
            Q(i,j) = -1/dt_hat(i+1)^2;
        elseif(i == j+1)&&(i≤N)
            Q(i,j) = -1/dt_hat(i)^2;
        end

        if(i==j) &&(i<2*N)
            A(i,j) = -1;
        elseif (j==i+N)&& (i≤N)
            A(i,j) = -1;
        elseif(j == i-N) && (i>N)&&(i<2*N)
            A(i,j) = 1;
        elseif(i==2*N) && (j ==N)
            A(i,j) = -1;
        elseif(i==2*N)&&(j==2*N+1)

```

```

        A(i,j) =-1;
elseif(i==2*N+1) && (j==N)
        A(i,j) =1;
elseif(i==2*N+1) && (j==2*N+1)
        A(i,j) =-1;
end
end

if(i≤N)
        b(i) = -SOC_min;
elseif(i>N) && (i<2*N)
        b(i) = SOC_max - dSOC_tilde(i-N);
elseif(i==2*N)
        b(i) = -SOC_set + dSOC_tilde(N)+ dSOC_f;
elseif(i==2*N+1)
        b(i) = SOC_set + dSOC_f - dSOC_tilde(N);
end

if(i==1) && (N==1)
        p(i) = - 2*SOC_set/dt_hat(N)^2;
elseif(i==1) && (N>1)
        p(i) = - 2*SOC_set/dt_hat(1)^2 + ...
                2*dSOC_tilde(1)/dt_hat(2)^2 ;
elseif(i≥2) && (i≤N-1) && (N>2)
        p(i) = -2*dSOC_tilde(i-1)/dt_hat(i)^2 + ...
                2*dSOC_tilde(i)/dt_hat(i+1)^2;
elseif (i==N) && (N>1)
        p(i) = -2*dSOC_tilde(N-1)/dt_hat(N)^2;
end

end

```

```

x = zeros((2*N+1),1,'double');
x = QP(2*Q,p,A,b);

t = zeros(2*N+1,1);

for j = 1:2*N+1
    if j==1
        SOC_ref(j) = SOC_set_mem;
        t(j) = 0;
        d(j) = veh_offset;
    elseif(mod(j,2)==0)
        SOC_ref(j) = x(j/2);
        t(j) = t_tilde(j/2);
        d(j) = dist_tilde(j/2);
    elseif(mod(j,2)==1)
        SOC_ref(j) = x((j-1)/2) + dSOC_tilde((j-1)/2);
        t(j) = t_tilde((j-1)/2) + dt_tilde((j-1)/2);
        d(j) = dist_tilde((j-1)/2) + ...
            del_dist_tilde((j-1)/2);
    end
end
c = 2*N+1;
d(c+1:end) =100*(1:1002-c)+d(c);
SOC_ref(c+1:end) = 0.55;
val_flag = 1>0;
else
    val_flag = 1<0;
end
flag = 0>1;
end
end

```

```

if(val_flag)
    temp = interp1(d,SOC_ref,veh_offset,'linear','extrap');
    SOC_setpt = temp(1);
else
    SOC_setpt = 0.55;
end
SOC_set_mem = SOC_setpt;
end

function y = QP(H,F,A,b)
coder.extrinsic('quadprog');
y = quadprog(H,F,A,b);
end

```

## REFERENCES

- [1] A. Sciarretta and L. Guzzella, *Control of hybrid electric vehicles*, 2007, IEEE Control Systems Magazine, Vol. 27, No. 2, pp. 60–70.
- [2] Hyeoun-Dong Lee, Seung-Ki Sul, *Fuzzy-logic-based torque control strategy for parallel-type hybrid electric vehicle*, 1998, IEEE Transactions on Industrial Electronics, Vol. 45, No. 4, pp. 625 - 632.
- [3] N. J. Schouten, M. A. Salman, N. A. Kheir, *Fuzzy logic control for parallel hybrid vehicles*, 2002, IEEE Transactions on Control Systems Technology, Vol.10, No.3, pp. 460 - 468.
- [4] NJ Schouten, MA Salman, NA Kheir, *Energy management strategies for parallel hybrid vehicles using fuzzy logic*, 2003, Control Engineering Practice, Vol. 11, No. 2, pp. 171–177.
- [5] G. Paganelli, S. Delprat, T. M. Guerra, J. Rimaux, *Equivalent consumption minimization strategy for parallel hybrid powertrains*, 2002, Vehicular Technology Conference, IEEE 55th, Vol.4, pp. 2076 - 2081.
- [6] Pinak Tulpule, Vincenzo Marano, Giorgio Rizzoni, *Energy management for plug-in hybrid electric vehicles using equivalent consumption minimisation strategy*, 2010, International Journal of Electric and Hybrid Vehicles, Vol. 2, No. 4.
- [7] Park, J. & Park, J., *Development of equivalent fuel consumption minimization strategy for hybrid electric vehicles*, 2012, International Journal of Automotive Technology, Vol. 13, No. 5, pp. 835–843.
- [8] Simona Onori, Lorenzo Serrao and Giorgio Rizzoni, *Adaptive equivalent consumption minimization strategy for hybrid electric vehicles*, 2010, ASME Dynamic Systems and Control Conference, Vol. 1, pp. 499-505.
- [9] Cristian Musardo, Giorgio Rizzoni, Yann Guezennec, Benedetto Staccia, *A-ECMS: An adaptive algorithm for hybrid electric vehicle energy management*, 2005, European Journal of Control, Vol. 11, No. 4, pp. 509-524.
- [10] Namwook Kim, Sukwon Cha, Huei Peng, *Optimal control of hybrid electric vehicles based on pontryagin's minimum principle*, 2010, IEEE Transactions on Control Systems Technology, Vol.19, No. 5, pp. 1279 - 1287.

- [11] Lorenzo Serrao, Simona Onori, Giorgio Rizzoni, *ECMS as a realization of Pontryagin's minimum principle for HEV control*, 2009, Proceedings of the American Control Conference, pp. 3964-3969.
- [12] M. Salman, M.-F. Chang, and J.-S. Chen, *Predictive energy management strategies for hybrid vehicles*, 2005, Proceedings of Vehicle Power and Propulsion, IEEE Conference, pp. 21-25.
- [13] Lars Johannesson, *Predictive control of hybrid electric vehicles on prescribed routes*, 2009, Chalmers University of Technology.
- [14] Daniel Ambuhl, Lino Guzzella, *Predictive reference signal generator for hybrid electric vehicles*, IEEE Transactions on Vehicular Technology, Vol. 58, No. 9, pp. 4730 - 4740.
- [15] Daniel Ambuhl, Olle Sundstrom, Antonio Sciarretta, Lino Guzzella, *Explicit optimal control policy and its practical application for hybrid electric powertrains*, 2010, Control Engineering Practice, Vol. 18, No. 12, pp. 1429-1439
- [16] Daniel Ambuhl, *Energy management strategies for hybrid electric vehicles*, 2009, ETH Zurich.
- [17] Mathworks Simulink, [http : //mathworks.com/products/simulink/?s\\_tid = srchtitle](http://mathworks.com/products/simulink/?s_tid=srchtitle).
- [18] Google Maps Elevation API, [https : //developers.google.com/maps/documentation](https://developers.google.com/maps/documentation).
- [19] Haversine Formula, [https : //en.wikipedia.org/wiki/Haversine\\_formula](https://en.wikipedia.org/wiki/Haversine_formula).
- [20] Arturo Locatelli, *Optimal control: An introduction*, 2001, Springer Science & Business Media.
- [21] Donald E. Kirk, *Optimal control theory: An introduction*, 2012, Courier Corporation.
- [22] Makiko Nisio, *Stochastic control theory: Dynamic programming principle*, 2014, Springer.
- [23] C. Musardo and B. Staccia, *Energy management strategies for hybrid electric vehicles*, 2003, Politecnico di Milano.
- [24] A. Sciarretta, L. Guzzella, and C. Onder, *On the power split control of parallel hybrid vehicles: from global optimization towards realtime control*, 2003, at-Automatisierungstechnik, Vol. 51, No. 5, pp. 195-203.

- [25] X. Wei, L. Guzzella, V. Utkin, and G. Rizzoni, *Model-based fuel optimal control of hybrid electric vehicle using variable structure control systems*, 2007, Journal of Dynamic Systems, Measurement and Control, Vol. 129, pp. 13–19.
- [26] J.J.P. van Boekel, I.J.M Besselink, H. Nijmeijer, *Design and realization of a One-Pedal-Driving algorithm for the TU/e Lupo EL*, 2015, EVS28, KINTEX, Korea.